Article

Personalized Standard Deviations Improve the Baseline Estimation of Collaborative Filtering Recommendation

Zhenhua Tan ^{1,*} , Liangliang He ², Danke Wu ¹, Qiuyun Chang ¹ and Bin Zhang ¹

¹ Software College, Northeastern University, Shenyang 110819, China; 1910467@stu.neu.edu.cn (D.W.); changqiuyun@stumail.neu.edu.cn (Q.C.); zhangbin@mail.neu.edu.cn (B.Z.)

² College of Computer, National University of Defense Technology, Changsha 410073, China; heliangliang19@nudt.edu.cn

* Correspondence: tanzh@mail.neu.edu.cn

Received: 9 June 2020; Accepted: 7 July 2020; Published: 10 July 2020

Featured Application: A novel baseline estimation model for collaborative filtering recommendation is proposed. Personalized standard deviations from global and local could improve the predictive accuracies.

Abstract: Baseline estimation is a critical component for latent factor-based collaborative filtering (CF) recommendations to obtain baseline predictions by evaluating global deviations for both users and items from personalized ratings. Classical baseline estimation presupposes that the user's factual rating range is the same as the system's given rating range. However, from observations on real datasets of movie recommender systems, we found that different users have different actual rating ranges, and users can be classified into four kinds according to their personalized rating criterion, including normal, strict, lenient, and middle. We analyzed ratings' distributions and found that the proportion of user ratings' local standard deviation to the system's global standard deviation is equal to that of the user's actual rating range to the system's rating range. We propose an improved and unified baseline estimation model based on the standard deviation's proportion to alleviate the influence of classical baseline estimation's limitation. We also apply the proposed baseline estimation model in existing latent factor-based CF recommendations and propose two instances. We performed experiments on full ratings of datasets by cross evaluations, including Flixster, Movielens (10 M), Movielens (latest small), FilmTrust, and MiniFilm. The results prove that the proposed baseline estimation model has better predictive accuracy than the classical model and is efficient in improving prediction performance for existing latent factor-based CF recommendations.

Keywords: recommender system; baseline estimation; collaborative filtering; standard deviation; personalization

1. Introduction

Recommender technologies have been applied to a variety of Internet-based systems, such as online multimedia services, online shopping, and online social networking. Collaborative filtering (CF) recommendation is one of the most important techniques for recommender systems [1,2]. The main techniques of CF recommendation include the neighborhood method and latent factor model [3]. The neighborhood method aims at improving predictive accuracy by similarities of relations, such as [4–7]. The latent factor model uses historical rating data to discover latent features in users' behaviors and

explain user ratings in conjunction with explicit evidence, such as [8–12], including baseline estimation model and latent factor model as in Equation (1):

$$\hat{r}_{ui} = (b_u + b_i + \mu) + f(q_i^T p_u). \quad (1)$$

The formalized function $f(q_i^T p_u)$ decomposes the user-item rating matrix R into users' latent factor matrix P and items' latent factor matrix Q , where P reflects the weight of users' preference on each latent factor, and Q reflects the weight of items' characteristics. Parameters q_i and p_u are related factors of Q and P . Most researchers have focused on algorithms of $f(q_i^T p_u)$ to obtain more accurate predictive accuracy by personalization methods, such as [5,8–14].

In this paper, we focus on the first part in Equation (1), called baseline estimation [5,8,14], where b_u and b_i denote the observed global deviations of user u and item i , respectively. The related equation is:

$$\hat{r}_{ui} = b_u + b_i + \mu, \quad (2)$$

where $\hat{r}_{ui}$ denotes the baseline prediction for an unknown item i by u , and μ denotes the global average rating of the system. Baseline estimation evaluates the global deviations of users' personalized ratings on items against global average ratings and the deviations of items' received ratings to the overall average [5]. Usually, b_u and b_i can be observed from the user-item rating matrix by calculating the related row average rating and column average rating, denoted by $\bar{r}_u$ and $\bar{r}_i$, respectively, and $b_u = (\bar{r}_u - \mu)$ and $b_i = (\bar{r}_i - \mu)$.

Our main motivation comes from the limitation of the baseline estimation, which is firstly mentioned by Koren in [5,8]. The main problem of baseline estimation in (2) is that system cannot differentiate users' personalized rating range from the system's rating range. Factually, different users usually have different rating criteria on the same items. Some have relatively higher ratings for items than others, while some have very strict rating criteria for the same item. As a result, users' actual rating range may differ from the system's rating range, and the baseline estimation may result in an irrational predictive value. We believe that the main limitation of classical baseline estimation is the lack of local personalization consideration. From observations on real recommendation datasets, we found that users in all datasets can be classified into four kinds according to their real rating ranges: Normal, strict, lenient, and middle. The normal users gave ratings within the system's range, but the strict users' rating range's supremum was less than that of the system's range, while the lenient users' rating range's infimum was larger than that of the system's, and the middle users' rating scale did not cover both the supremum and infimum of the system's rating range. The observation proves that users' real rating range is influenced by users' personalized rating criteria, and baseline estimation should not only consider global deviation from the system's rating range but also consider the local deviation from users' personalized rating ranges. We discuss this problem in Section 3 in detail.

Our main purpose was to propose an improved and unified baseline estimation model from local deviations and global deviations. We investigated local deviations from different kinds of users' rating distributions and found a standard deviation proportion (SDP) pattern for baseline estimation. To our knowledge, this study is the first to improve the baseline estimation from the perspective of a normal distribution. This work has the following main contributions:

- (1) Observed that four kinds of personalized users exist in recommender systems with personalized rating criteria;
- (2) Proposed an improved and unified baseline estimation model based on users' personalized rating distributions and system ratings' global distribution; and
- (3) Proposed application instances of SDP to existing latent factor-based CF recommendations with excellent improved predictive accuracies.

Since the standard deviations can be calculated from historical data, the proposed SDP has no additional cost compared to classical baseline estimation in real systems. Experiments on real datasets

with full ratings and cross validation prove that the proposed SDP is more efficient than classical baseline estimation and can effectively improve the predictive accuracies of existing latent factor-based CF recommendations. The rest of this article is organized as follows. We introduce the related work in Section 2 and problem statements in Section 3. We describe SDP in Sections 4 and 5, and experiments and evaluation in Section 6, followed by conclusions in Section 7.

2. Related Work

Recommendation techniques mainly include content-based recommendations, collaborative filtering (CF), and hybrid recommendations. Content-based recommendation focuses on user profiles and preferences [15–17], while CF recommendation focuses on user ratings related to the user-item matrix to find a set of like-minded users or similar items [18,19], and hybrid recommendation combines two or more recommendation techniques [20].

Collaborative filtering (CF) techniques play a significant role in recommender systems and can be classified into neighborhood-based CF and latent factor model-based CF. Neighborhood-based CF focuses on user ratings related to the user-item matrix to find a set of likeminded users or similar items and is mainly classified into user-based CF and item-based CF [21]. User-based CF discovers users with similar interests or preferences on items to a given user based on users' similarities, while item-based CF usually recommends similar items to a user based on items' correlations [22–24]. Components of reliability [6,7], users' relationships and interest [22], personalized behaviors [23], and similarities [2,24] are often used to improve the predictive accuracy of neighborhood-based CF recommendations.

Personalization is the core topic during recommendation [25]. Latent factor-based CF uses historical rating data to discover latent features in users' behaviors. Because latent factor-based CF can train and learn users' personalization offline, it has higher prediction and has become one of the most popular topics in recommender systems since the Netflix Prize competition. The latent factor model is derived from the singular value decomposition (SVD) [26]. Koren investigated baseline estimates, neighborhood models, and latent factor models of CF recommendations and proposed SVD++ by extending SVD to make full use of implicit feedback information [5]. Kumar et al. [9] incorporated social popularity factors in the SVD++ matrix factorization model to improve the practical accuracy of recommendations. Bao et al. [10] adopted topic modeling to model latent topics in review content and used biased matrix factorization (MF) for prediction in recommender systems, simultaneously correlating with user and item factors. Guo et al. [11,12] investigated the effectiveness of novel fusing of trust networks to rating prediction, where not only the explicit but also the implicit influence of trust are integrated into the SVD model. Pan et al. [13] proposed a self-transfer learning algorithm to iteratively identify and integrate some likely positive unlabeled feedback into the learning task of labeled feedback. In the past five years, deep learning has also been imported into the recommendation to discover more useful latent factors based on users' preferences by big data analysis. Wu et al. [27] proposed a deep latent factor model based on a high-dimensional and sparse matrix to construct a deep-structured model by sequentially connecting multiple latent factor models. Li et al. [28] proposed a combined deep CF framework by probabilistic matrix factorization with marginalized denoising stacked autoencoders, deep learning the effective latent representations and achieving good performance. Zhang et al. [29] summarized related research efforts on deep learning-based recommender systems with expanded trend analysis and mentioned that recommendations need better, more unified, and harder evaluation. As argued in [1], the assessment criterion should be considered when choosing the appropriate recommendation.

The baseline estimation model is a critical component for latent factor model-based CF recommendations, such as in [3,5,8,9,11–14]. Baseline estimation can evaluate the global deviations of users' personalized ratings on items against global average ratings and the deviations of items' received ratings to the overall average [5]. The important elements in baseline estimation are b_u , b_i , and μ . In contrast to most researchers that focused on modeling the latent factors, fewer researchers found that the baseline estimation could also consider more factors to enhance the global deviation predictions. In [8], Koren

observed that users' preferences for items drifted over time, and items' popularity also constantly change. He modeled the dynamics over time and proposed a time factor-based CF recommendation (timeSVD++), where the baseline estimations were modeled as $\hat{b}_{ui} = \mu + b_u(t) + b_i(t)$. Tan et al. [14] proposed a proportion-based baseline estimation model for CF recommendation by considering personalized rating segments, and deviations b_i were weighted with factor $(b_u + \mu - m)/(\mu - m)$ when b_u is negative, and with $(n - b_u - \mu)/(n - \mu)$ when b_u is positive, where $[m, n]$ is the rating segment.

The Gaussian process is usually applied to improve the accuracy of recommendations. Zhang et al. [16] proposed a modified EM model by Bayesian hierarchical linear models assuming user models are sampled randomly from a Gaussian distribution. Sofiane and Bermak [30] proposed a Bayesian procedure based on the Gaussian process using a nonstationary covariance function for time series prediction. Ko et al. [31] integrated Gaussian processes into different forms of Bayes filters; both considered the global mean of previous states and related uncertainty.

Inspired by the above, our motivation was to find an improved and unified baseline estimation model for latent factor-based CF recommendation, with consideration of both global deviations and local deviations from users' personalized normal distributions. A preliminary version of this report appeared in the Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME) [32].

3. Problem Statement

In the baseline estimation model in (2), related deviations b_u and b_i are both based on the system's global rating range, and it has inherent limitations in controlling the predictive value from overflowing the system's rating range. This global deviation may result in irrational predictive results, and the prediction may even overflow the rating range. For example, assume that the entire recommender system's average rating is $\mu = 4.0$, and the system's rating range is from 0.5 to 5.0. For an unromantic movie M, it obtains an average rating $\bar{r}_i = 2.0$, so the related global deviation of movie M would be $b_i = 2.0 - 4.0 = -2.0$. Alice is a critic and has a very strict rating criterion on movies, and her average rating on movies is $\bar{r}_u = 1.5$, so the related global deviation of Alice would be $b_u = 1.5 - 4.0 = -2.5$. The possible baseline estimate on movie M by Alice would be $\hat{r}_{ui} = (-2.0) + (-2.5) + 4.0 = -0.5 < 0$. Obviously, this predictive value is irrational because it does not meet the ranges. More examples were demonstrated in [14,32].

In real systems, users always give ratings according to their personalized rating criterion; some are strict, and some are lenient. Ideally, the baseline estimation will be accurate if we fully consider each user's personalization. However, it is time costly and unrealistic in real systems. Therefore, we focused on how to discover a unified personalized factor with commonality to improve the baseline estimation predictive accuracy and attempted to apply it to improve the performance of related CF recommendation algorithms, considering not only global deviation from system ratings but also local deviation from users' personalized ratings.

We observed that users in many recommender systems can be classified into four kinds by their personalized rating ranges.

- (1) *Normal user*. The actual rating range of this kind of user usually covers the recommender system's rating range. For example, assume that the system's rating range is between [1,5], those normal users' rating range may be the same as [1,5].
- (2) *Strict user*. In this situation, users have relatively strict criteria and make a relatively lower rating than others. They rate items strictly, and few cover the highest rating range.
- (3) *Lenient user*. These users habitually give relatively higher ratings than others, and their actual rating range usually does not cover the lowest range.
- (4) *Middle user*. Users of this kind neither give higher ratings nor give lower ratings, and their actual rating range is the middle, e.g., [2,4] while the system's range is [1,5].

We performed rating statistics over five real datasets, including Flixster, MiniFilm, FilmTrust, Movielens (10 M), and Movielens (Latest Small), to verify the existence of these four kinds of

personalized rating behaviors. Figure 1 shows the rating statistics. During the observation, we defined that the normal users' ratings cover all the system's rating scale, the strict users' ratings do not cover the highest boundary, and the lenient users' ratings do not cover the lowest, while the middle users' ratings do not include both the highest boundary and lowest boundary. From the statistics, we can see that the four situations exist in all datasets. The observation shows that most users were lenient or normal, some were middle, while a few users were strict, and proves that the real rating range of users may be less than the system's rating range. Therefore, it is necessary to consider local deviation during the baseline estimation.

Figure 1. Rating statistics according to four kinds of personalized rating behavior: normal, strict, lenient, and middle on the Flixster, MiniFilm, FilmTrust, MovieLens (10 M), and MovieLens (Latest Small) datasets.

4. Proposed Model

The main purpose of this paper was to discover a more rational baseline estimation model from both local deviations and global deviations to improve predictive accuracy. This section will introduce the processes of the proposed improved baseline estimation model.

4.1. Symbols Definition

Definition 1. Related symbols refer to Table 1.

Definition 2. Let Ran_{sys} denote the system's rating range, and Ran_u be the actual rating range of user u . For example, Ran_{sys} of a recommender system may be 0.5 to 5.0, and Ran_u of a middle user may be 1.0 to 4.0.

Definition 3. Let Len_{sys} and Len_u denote the length of $Range_{sys}$ and $Range_u$, respectively, calculated by the highest bound minus the lowest bound of the related rating range. Let us take examples based on the above; then, $Len_{sys} = 5.0 - 0.5 = 4.5$, and $Len_u = 4.0 - 1.0 = 3.0$.

Table 1. Some related symbols.

Symbols	Description
u, v	user ID
i, j	item ID
μ	overall average rating
μ_u	average rating given from u
$\bar{r}_i$	average rating of i
r_{ui}	observed rating of (u, i)
$\hat{r}_{ui}$	predicted rating of (u, i)
b_u	bias of u
b_i	bias of i
σ_u	Local standard deviation of u
σ	Global standard deviation
N_{sys}	The number of all ratings in system
N_u	The number of ratings given by user u
$Q \in \mathbb{R}^{d \times m}$	factor-item matrix
$p_u \in \mathbb{R}^{1 \times d}$	latent feature vector of user u
$q_i \in \mathbb{R}^{1 \times d}$	latent feature vector of item i

4.2. Observation from Rating Distribution

(1) Global observation

As argued in many related works [16,31,32], the overall ratings follow the normal distribution in recommender systems when there are enough rating behaviors. Theoretically, the rating densities follow $\varphi(y) = N(\mu, \sigma^2)$, with the probability density as:

$$\varphi(y) = N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \quad (3)$$

where $y \in [\mu - \frac{Len_{sys}}{2}, \mu + \frac{Len_{sys}}{2}]$, μ is the mean value of whole ratings in the system, σ^2 is the global variance in all ratings, and Len_{sys} is the length of the system's rating scale. Additionally:

$$\sigma = \frac{1}{N_{sys}} \sum_{n=1}^{N_{sys}} (r_n - \mu), \quad (4)$$

represents the global standard deviation of ratings, where N_{sys} is the number of all ratings described.

We selected five real datasets for the global observation, including large-scale datasets Flixster and Movielens-10 M, middle-scale dataset FilmTrust, small-scale dataset ml-latest-small, and mini-scale dataset MiniFilm. The related properties of these datasets are described in Section 6. Figure 2 shows the rating densities' distribution with different rating segments. We observed that these ratings follow the general normal distribution trend, especially when there are more ratings. This observation proves that the normal distribution probability can adapt to the global distribution for overall ratings in all recommender systems.

Figure 2. Global distributions on different datasets: (a) Flixster, (b) MiniFilm, (c) FilmTrust, (d) MovieLens_10 M, and (e) MovieLens_Latest_Small.

(2) Local observation

To investigate the personalized rating features, we further observed the local distribution of ratings given from different kinds of personalized users. We selected Flixster for the example demonstration to make the description more convenient. Other datasets with enough ratings also follow a similar regularity. As described in Section 2, we found that four types of personalized users exist in recommender systems, including normal users, strict users, middle users, and lenient users. We calculated the statistics on all 8,196,077 ratings of Flixster according to different users' rating segments. Table 2 shows the statistical results. The rating range of normal users is from 0 to 5, that of strict users is from 0 to 4, that of middle users is from 1 to 4, and that of lenient users is from 1 to 5. We also investigated related segments' rating densities and calculated their mean values μ_u and standard deviations σ_u . Meanwhile, we calculated the fitted mean value μ_f and fitted standard deviation σ_f for different kinds of users' ratings according to a normal distribution.

Table 2. Statistics of the rating distribution of different kinds of users in Flixster.

User Type	Rating Range	#Related Ratings	Densities	Statistical Parameters		Fitted Parameters	
				μ_u	σ_u	μ_f	σ_f
<i>Normal</i>	[0,1)	141,610	0.0280	3.4506	1.1529	3.452	1.752
	[1,2)	322,168	0.0638				
	[2,3)	663,944	0.1315				
	[3,4)	1,719,123	0.3405				
	[4,5)	1,366,972	0.2707				
	[5,6)	835,220	0.1654				
<i>Strict</i>	[0,1)	12,135	0.1042	2.6286	1.1321	2.788	1.52
	[1,2)	15,200	0.1305				
	[2,3)	24,156	0.2074				
	[3,4)	47,324	0.4063				
	[4,5)	17,649	0.1515				
	[5,6)	–	0.0000				
<i>Lenient</i>	[0,1)	–	0.0000	3.9853	0.8665	3.991	1.689
	[1,2)	38,980	0.0142				
	[2,3)	156,224	0.0570				
	[3,4)	836,237	0.3054				
	[4,5)	950,718	0.3472				
	[5,6)	756,329	0.2762				
<i>Middle</i>	[0,1)	–	0.0000	3.3091	0.8353	3.251	1.154
	[1,2)	17,501	0.0599				
	[2,3)	41,515	0.1421				
	[3,4)	138,739	0.4750				
	[4,5)	94,333	0.3230				
	[5,6)	–	0.0000				

We demonstrate the related rating densities in stem style in Figure 3 to observe the local distribution of ratings given by different kinds of users. We observed that the rating densities of all kinds of users generally follow normal distributions with different deviations, as shown in stem Figure 3. Intuitively, we fitted these densities into normal distributions based on (μ_u, σ_u) , as shown in the dashed curves in Figure 3. Additionally, for each kind of user, we fitted the complete rating densities into a normal distribution based on the related mean (μ_f, σ_f) , as shown in the blue curves in Figure 3. We found that the statistical distribution of $N(\mu_u, \sigma_u^2)$ is close to the fitted $N(\mu_f, \sigma_f^2)$. The main differences between them on the same ratings remain in the related standard deviation and related mean value. From the observation, we found that the rating distribution of different kinds of users can also follow the normal distribution. We call this a local distribution in this paper and define that ratings given by a personalized user u follow the normal distribution $\varphi_u(x) = N(\mu_u, \sigma_u^2)$, with the probability density as:

$$\varphi_u(x) = N(\mu_u, \sigma_u^2) = \frac{1}{\sqrt{2\pi}\sigma_u} e^{-\frac{(x-\mu_u)^2}{2\sigma_u^2}}, \quad (5)$$

where $x \in [\mu_u - \frac{Len_u}{2}, \mu_u + \frac{Len_u}{2}]$ and μ_u is the local mean value by u and σ_u is the related local standard deviation.

Figure 3. Local distribution of ratings from different kinds of users on Flixster. Rating densities in stem style, fitted curves of $N(\mu_u, \sigma_u^2)$ and $N(\mu_f, \sigma_f^2)$.

(3) Guess and Proposition

The main purpose of this paper was to find a more rational estimation model based on the global distribution $\varphi(y) = N(\mu, \sigma^2)$ and local distribution $\varphi_u(x) = N(\mu_u, \sigma_u^2)$. Observation proves that different personalized rating ranges result in different standard deviations but still follow local normal distributions. In our opinion, with ratings increasing, the cumulative distribution of different kinds of users' ratings will be closer and the same when there are enough rating behaviors. To obtain a relatively generic result, we assumed that the cumulative distribution of a specific user's normal distribution is equal to that of the system's normal distribution, and we made the following hypothesis according to different rating ranges and distributions.

Hypothesis 1 (H1). In a RecSys, the cumulative distributions of $\varphi_u(x)$ and $\varphi(y)$ constrained to related rating ranges remain the same, meaning that the coverage area of the local normal distribution $\varphi_u(x)$ between $[\mu_u - \frac{Len_u}{2}, \mu_u + \frac{Len_u}{2}]$ is equal to the coverage of the global distribution $\varphi(y)$ between $[\mu - \frac{Len_{sys}}{2}, \mu + \frac{Len_{sys}}{2}]$; that is:

$$\Phi_u\left(x \in \left[\mu_u - \frac{Len_u}{2}, \mu_u + \frac{Len_u}{2}\right]\right) = \Phi\left(y \in \left[\mu - \frac{Len_{sys}}{2}, \mu + \frac{Len_{sys}}{2}\right]\right) = K, \quad (6)$$

where K is a constant and less than 1.0.

Based on this hypothesis, we focused on the standard deviations and personalized rating ranges and propose the following.

Proposition 1. The proportion of $\left(\frac{\sigma_u}{\sigma}\right)$ is equal to $\left(\frac{Len_u}{Len_{sys}}\right)$, that is $\left(\frac{\sigma_u}{\sigma}\right) = \left(\frac{Len_u}{Len_{sys}}\right)$.

Proof. The cumulative distribution of $\varphi_u(x)$ is:

$$\Phi_u(x) = \int_{\mu_u - \frac{Len_u}{2}}^{\mu_u + \frac{Len_u}{2}} \varphi_u(x) dx = \int_{\mu_u - \frac{Len_u}{2}}^{\mu_u + \frac{Len_u}{2}} \frac{1}{\sqrt{2\pi}\sigma_u} e^{-\frac{(x-\mu_u)^2}{2\sigma_u^2}} dx.$$

Let $\left(\frac{x-\mu_u}{\sigma_u}\right) = t$, then:

$$\Phi_u(x) = \int_{-\frac{Len_u}{2\sigma_u}}^{\frac{Len_u}{2\sigma_u}} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \frac{2}{\sqrt{2\pi}} \int_0^{\frac{Len_u}{2\sigma_u}} e^{-\frac{t^2}{2}} dt.$$

The cumulative distribution of $\varphi(y)$ is:

$$\Phi(y) = \int_{\mu - \frac{Len_{sys}}{2}}^{\mu + \frac{Len_{sys}}{2}} \varphi(y) dy = \int_{\mu - \frac{Len_{sys}}{2}}^{\mu + \frac{Len_{sys}}{2}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}} dy.$$

Let $\left(\frac{y-\mu}{\sigma}\right) = t$, then:

$$\Phi(y) = \int_{-\frac{Len_{sys}}{2\sigma}}^{\frac{Len_{sys}}{2\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \frac{2}{\sqrt{2\pi}} \int_0^{\frac{Len_{sys}}{2\sigma}} e^{-\frac{t^2}{2}} dt.$$

According to the hypothesis, $\Phi(y) = \Phi_u(x) = K$, then:

$$\frac{2}{\sqrt{2\pi}} \int_0^{\frac{Len_{sys}}{2\sigma}} e^{-\frac{t^2}{2}} dt = \frac{2}{\sqrt{2\pi}} \int_0^{\frac{Len_u}{2\sigma_u}} e^{-\frac{t^2}{2}} dt = K.$$

Thus, the proportion of $\left(\frac{Len_{sys}}{2\sigma}\right)$ is equal to the proportion of $\left(\frac{Len_u}{2\sigma_u}\right)$; that is, $\left(\frac{Len_{sys}}{2\sigma}\right) = \left(\frac{Len_u}{2\sigma_u}\right)$.

Thus, we obtained the following relation:

$$\left(\frac{\sigma_u}{\sigma}\right) = \left(\frac{Len_u}{Len_{sys}}\right).$$

□

4.3. Proposed Improved Baseline Estimation Model

Based on the above, we conclude the improved baseline estimation model based on the proportion of local standard deviation to global standard deviation. According to $\left(\frac{\sigma_u}{\sigma}\right) = \left(\frac{Len_u}{Len_{sys}}\right)$, we can obtain:

$$\frac{(\hat{r}_{ui} - \mu_u)}{(\bar{r}_i - \mu)} = \left(\frac{\sigma_u}{\sigma}\right), \quad (7)$$

$$\hat{r}_{ui} = \frac{\sigma_u}{\sigma} \cdot (\bar{r}_i - \mu) + \mu_u. \quad (8)$$

Since b_u and b_i are the observed deviations of user u and item i , respectively, generally $b_u = \bar{r}_u - \mu$, $b_i = \bar{r}_i - \mu$, so:

$$\hat{r}_{ui} = \left(\frac{\sigma_u}{\sigma}\right) \cdot b_i + b_u + \mu. \quad (9)$$

This is the improved baseline estimation with the proportion of the global standard deviation to the local standard deviation, and we call this novel model SDP (standard deviation proportion baseline estimate). For parameters b_u and b_i , we can use the following loss function to obtain the optimized b_u^* and b_i^* :

$$\mathcal{L}_{SDP} = \min_{b_u^*, b_i^*} \sum_{(u,i,r) \in \mathbb{R}} \left(r_{ui} - \left(\frac{\sigma_u}{\sigma}\right) b_i - b_u - \mu \right)^2. \quad (10)$$

Since the local standard deviation σ_u is usually less than the global standard deviation σ , the above SDP can satisfy most situations. However, the sparsity problem appears when the number of ratings is less than 20, where standard deviation does not follow the normal distribution. Thus, we adjusted the SDP into the following:

$$\hat{r}_{ui} = \begin{cases} \frac{\sigma_u}{\sigma} \cdot b_i + b_u + \mu, & \text{if } \sigma_u \leq \sigma \\ b_i + b_u + \mu, & \text{if } \sigma_u > \sigma \end{cases}. \quad (11)$$

This means that the SDP uses the original baseline estimation with no features of a normal distribution to limit the influence of the sparsity problem when $\sigma_u > \sigma$. In the next section, we introduce how to apply SDP into existing latent factor-based recommendation algorithms.

5. Application Instances of Proposed SDP

In this section, we provide two instances to illustrate how to apply the proposed SDP baseline estimation model to existing latent factor-based recommendation algorithms that use classical baseline estimation. The main method adds factor $\left(\frac{\sigma_u}{\sigma}\right)$ to b_i and makes some improvement during model learning. We selected two famous and efficient existing latent factor-based CF recommendations for improved instances. The first instance improves SVD++ [5] by SDP, named SDPSVD++; the second improves TrustSVD [11,12] by SDP, named SDPTrustSVD.

5.1. Application Instance-1: SDPSVD++

(1) Improved by SDP

SVD++ extends SVD using feedback from user ratings. The detailed improvement is that a free factor-user 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:

$$\hat{r}_{ui} = b_u + b_i + \mu + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j\right) \right), \quad (12)$$

where I_u represents the set of item ratings by user u ; $y_j \in Y$ represents the implicit influence of items rated by user u in the past on the ratings of unknown items in the future.

Based on the SDP model, we improved the baseline estimation using a proportion of $\left(\frac{\sigma_u}{\sigma}\right)$ for SVD++, leading to the following model:

$$\hat{r}_{ui} = \left(\frac{\sigma_u}{\sigma}\right) \cdot b_i + b_u + \mu + q_i^T \cdot \left(p_u + |I_u|^{-\frac{1}{2}} t \sum_{j \in I_u} y_j\right). \quad (13)$$

All of these parameters have the same meanings as in SVD++. We named the new SVD++ model (13) SDPSVD++.

(2) Model learning

The involved parameters in Equation (13) of SDPSVD++ are learned by minimizing the regularized squared error function associated with (14):

$$\begin{aligned} \mathcal{L} = & \min_{q^*, p^*, y^*, b^*, \sigma_u^*} \left(\frac{1}{2} \sum_u \sum_{i \in I_u} \left(\hat{r}_{ui} - \left(\frac{\sigma_u}{\sigma}\right) \cdot b_i - b_u - \mu \right)^2 \right. \\ & + \frac{\lambda}{2} \sum_u |I_u|^{-\frac{1}{2}} (b_u^2 + \sigma_u^2 + \|p_u\|_F^2) + \\ & \left. \frac{\lambda}{2} \sum_i |U_i|^{-\frac{1}{2}} (b_i^2 + \|q_i\|_F^2) + \frac{\lambda}{2} \sum_j |U_j|^{-\frac{1}{2}} \|y_j\|_F^2 \right), \end{aligned} \quad (14)$$

where U_i and U_j are the set of users who rate items j and i , respectively; $\|\cdot\|$ denotes the Frobenius norm; and λ alleviates operational complexity and avoids overfitting. To obtain a local minimization of the objective function $\mathcal{L}$, we performed the following gradient descents on $b_u, b_i, \sigma_u, p_u, q_i$, and y_j for all the users and items in each given training dataset:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial b_u} &= \sum_{i \in I_u} e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot b_u \\ \frac{\partial \mathcal{L}}{\partial b_i} &= \sum_{u \in U_i} \left(\frac{\sigma_u}{\sigma}\right) e_{ui} + \lambda |U_i|^{-\frac{1}{2}} \cdot b_i \\ \frac{\partial \mathcal{L}}{\partial \sigma_u} &= \sum_{i \in I_u} \left(\frac{b_i}{\sigma}\right) e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot \sigma_u \\ \frac{\partial \mathcal{L}}{\partial p_u} &= \sum_{i \in I_u} e_{ui} q_i + \lambda |I_u|^{-\frac{1}{2}} p_u \\ \frac{\partial \mathcal{L}}{\partial q_i} &= \sum_{u \in U_i} e_{ui} (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j) + \lambda |U_i|^{-\frac{1}{2}} q_i \\ \forall j \in I_u, \frac{\partial \mathcal{L}}{\partial y_j} &= \sum_{i \in I_u} e_{ui} |I_u|^{-\frac{1}{2}} q_i + \lambda |U_j|^{-\frac{1}{2}} y_j. \end{aligned}$$

During the above gradient descent, there is $e_{ui} = \hat{r}_{ui} - r_{ui}$. Processes of $\frac{\partial \mathcal{L}}{\partial b_u}, \frac{\partial \mathcal{L}}{\partial p_u}, \frac{\partial \mathcal{L}}{\partial q_i}$, and $\frac{\partial \mathcal{L}}{\partial y_j}$ are the same as those in SVD++, while $\frac{\partial \mathcal{L}}{\partial b_i}$ adds factor $\left(\frac{\sigma_u}{\sigma}\right)$ to e_{ui} , and $\frac{\partial \mathcal{L}}{\partial \sigma_u}$ is added during gradient descent according to $\mathcal{L}_{SDP}$ in (10). Finally, latent factor matrices for the user and item, P and Q , are output when the loss function $\mathcal{L}$ reaches a local minimization.

Algorithm 1 shows how to obtain the prediction $\hat{r}_{ui}$ by learning $\mathcal{L}$. Given a dataset, we input the user-item matrix. The initial learning rate γ_1 is for $\frac{\partial \mathcal{L}}{\partial b_u}, \frac{\partial \mathcal{L}}{\partial b_i}$, and $\frac{\partial \mathcal{L}}{\partial \sigma_u}$, while rate γ_2 for $\frac{\partial \mathcal{L}}{\partial p_u}, \frac{\partial \mathcal{L}}{\partial q_i}$, and $\frac{\partial \mathcal{L}}{\partial y_j}$. A very small positive value ε (line 1, line 6) is to control the learning convergence of $\mathcal{L}$. According to the SDP model in Equations (11) and (13), we calculated the prediction of SDPSVD++ while $\sigma_u \leq \sigma$, as described between line 4 to line 14. If $\sigma_u > \sigma$, the prediction would follow the original learning algorithm in SVD++ according to Equations (11) and (12). The time complexity of $\mathcal{L}$ is $O(d|\mathbb{R}|)$, where d represents the matrix dimensions, and $|\mathbb{R}|$ represents the number of the observed ratings.

Algorithm 1: Get Prediction $\hat{r}_{ui}$ in SDPSVD++ by Learning $\mathcal{L}$ **Input:** user-item rating matrix $\mathbb{R}$, λ , learning rate γ_1 and γ_2 **Output:** Rating prediction $\hat{r}_{ui}$

- (1) Initialize ε with very small positive value (as 0.0001);
- (2) Calculate global deviation σ under Equation (3);
- (3) Calculate local standard deviation σ_u of user u under Equation (5);
- (4) **if** $\sigma_u \leq \sigma$ **then**
- (5) Begin learning with loss function $\mathcal{L}$ according to Equation (14);
- (6) **while** $e_{ui} > \varepsilon$ **do**
- (7) $b_u \leftarrow b_u - \gamma_1 \left(\sum_{i \in I_u} e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot b_u \right)$;
- (8) $b_i \leftarrow b_i - \gamma_1 \left(\sum_{u \in U_i} \left(\frac{\sigma_u}{\sigma} \right) \cdot e_{ui} + \lambda |U_i|^{-\frac{1}{2}} \cdot b_i \right)$;
- (9) $\sigma_u \leftarrow \sigma_u - \gamma_1 \left(\sum_{i \in I_u} \left(\frac{b_i}{\sigma} \right) e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot \sigma_u \right)$;
- (10) $p_u \leftarrow p_u - \gamma_2 \left(\sum_{i \in I_u} e_{ui} q_i + \lambda |I_u|^{-\frac{1}{2}} p_u \right)$;
- (11) $q_i \leftarrow q_i - \gamma_2 \left(\sum_{u \in U_i} e_{ui} (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j) + \lambda |U_i|^{-\frac{1}{2}} q_i \right)$;
- (12) $\forall j \in I_u, y_j \leftarrow y_j - \gamma_2 \left(\sum_{i \in I_u} e_{ui} |I_u|^{-\frac{1}{2}} q_i + \lambda |U_j|^{-\frac{1}{2}} y_j \right)$;
- (13) **end while**
- (14) Calculate $\hat{r}_{ui}$ according to Equation (13) of SDPSVD++
- (15) **else**
- (16) Begin learning with loss function according to algorithm in SVD++;
- (17) Calculate $\hat{r}_{ui}$ according to Equation (12) of SVD++;
- (18) **end if**
- (19) **return** $\hat{r}_{ui}$

5.2. Application Instance-2: SDPTrustSVD

(1) Improved by SDP

The second instance improves TrustSVD [10] based on SDP. The TrustSVD considers explicit users' trust information in the SVD++ [5] and merges the trust factor to obtain better accuracy. The equation is as follows:

$$\hat{r}_{ui} = b_u + b_i + \mu + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \left(\sum_{j \in I_u} y_j \right) + |T_u|^{-\frac{1}{2}} \cdot \left(\sum_{v \in T_u} w_v \right) \right), \quad (15)$$

where w_v denotes the latent factor vector of users (trustees) trusted by user u ; T_u denotes the set of users trusted by u . To apply the proposed SDP, we add the proportion $\left(\frac{\sigma_u}{\sigma} \right)$ to b_i :

$$\hat{r}_{ui} = \left(\frac{\sigma_u}{\sigma} \right) t b_i + b_u + \mu + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \cdot \sum_{j \in I_u} y_j + |T_u|^{-\frac{1}{2}} \cdot \sum_{v \in T_u} w_v \right). \quad (16)$$

All other parameters in the improved equation have the same meaning as the original TrustSVD, and we named this improved TrustSVD as SDPTrustSVD.

(2) Model learning

Involved parameters in SDPTrustSVD are learned by minimizing the regularized squared error function association. The regularized squared error function $\mathcal{L}$ is:

$$\begin{aligned} \mathcal{L} = & \min_{q^*, p^*, y^*, b^*, w^*, \sigma_u^*} \left(\frac{1}{2} \sum_u \sum_{i \in I_u} (\hat{r}_{ui} - r_{ui})^2 + \frac{\lambda_t}{2} \sum_u \sum_{v \in T_u} (\hat{t}_{uv} - t_{uv})^2 + \right. \\ & \frac{\lambda}{2} \sum_u |I_u|^{-\frac{1}{2}} (b_u^2 + \sigma_u^2) + \frac{\lambda}{2} \sum_i |U_i|^{-\frac{1}{2}} (b_i^2 + \|q_i\|_F^2) + \sum_u \left(\frac{\lambda}{2} |I_u|^{-\frac{1}{2}} + \frac{\lambda_t}{2} |T_u|^{-\frac{1}{2}} \right) \|p_u\|_F^2 + \\ & \left. \frac{\lambda}{2} \sum_j |U_j|^{-\frac{1}{2}} \|y_j\|_F^2 + \frac{\lambda}{2} |T_v|^{-\frac{1}{2}} \|w_v\|_F^2 \right), \end{aligned} \quad (17)$$

where $\hat{t}_{uv}$ denotes the predictive trust value of user v from user u , t_{uv} is the observed trust value, w_v denotes a user-specific latent feature vector of users (trustees) trusted by user u , T_u denotes the set of users trusted by user u , T_v^+ denotes the set of users who trust user v , and λ_t is a parameter to control the degree of trust regularization. To obtain a local minimization of the objective function $\mathcal{L}$, we performed the gradient descents on $b_u, b_i, \sigma_u, p_u, q_i, y_j$, and w_v for all the users and items in a given training dataset:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial b_u} &= \sum_{i \in I_u} e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot b_u \\ \frac{\partial \mathcal{L}}{\partial b_i} &= \sum_{u \in U_i} \left(\frac{\sigma_u}{\sigma} \right) \cdot e_{ui} + \lambda |U_i|^{-\frac{1}{2}} \cdot b_i \\ \frac{\partial \mathcal{L}}{\partial \sigma_u} &= \sum_{i \in I_u} \left(\frac{b_i}{\sigma} \right) e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot \sigma_u \\ \frac{\partial \mathcal{L}}{\partial p_u} &= \sum_{i \in I_u} e_{u,i} q_i + \lambda_t \sum_{v \in T_u} e_{u,v} w_v + \left(\lambda |I_u|^{-\frac{1}{2}} + \lambda_t |T_u|^{-\frac{1}{2}} \right) p_u \\ \frac{\partial \mathcal{L}}{\partial q_i} &= \sum_{u \in U_i} e_{u,i} (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j + |T_u|^{-\frac{1}{2}} \sum_{v \in T_u} w_v) + \lambda |U_i|^{-\frac{1}{2}} q_i \\ \forall j \in I_u, \frac{\partial \mathcal{L}}{\partial y_j} &= \sum_{i \in I_u} e_{ui} |I_u|^{-\frac{1}{2}} q_i + \lambda |U_j|^{-\frac{1}{2}} y_j \\ \forall v \in T_u, \frac{\partial \mathcal{L}}{\partial w_v} &= \sum_{i \in I_u} e_{u,i} |T_u|^{-\frac{1}{2}} q_i + \lambda_t e_{u,v} p_u + \lambda |T_v^+|^{-\frac{1}{2}} w_v.\end{aligned}$$

During the above process, e_{ui} is also equal to $(\hat{r}_{ui} - r_{ui})$, and $\frac{\partial \mathcal{L}}{\partial b_u}, \frac{\partial \mathcal{L}}{\partial p_u}, \frac{\partial \mathcal{L}}{\partial q_i}, \frac{\partial \mathcal{L}}{\partial y_j}$, and $\frac{\partial \mathcal{L}}{\partial w_v}$ are the same as the related processes in TrustSVD, and $\frac{\partial \mathcal{L}}{\partial b_i}$ is renewed by factor $\left(\frac{\sigma_u}{\sigma} \right)$, while $\frac{\partial \mathcal{L}}{\partial \sigma_u}$ is calculated during SDPTrustSVD according to $\mathcal{L}_{SDP}$ in (10). As a result, with the minimization convergence of the error function $\mathcal{L}$, latent factor matrices for the user and item, P and Q , are output. Algorithm 2 with pseudocode shows the learning processes during predicting $\hat{r}_{ui}$.

Algorithm 2: Get Prediction $\hat{r}_{ui}$ in SDPTrustSVD by Learning $\mathcal{L}$

Input: user-item rating matrix $\mathbb{R}$, user-user Trust matrix T , regularization parameters λ and λ_t , learning rate γ_1 and γ_2

Output: Rating prediction $\hat{r}_{ui}$

- (1) Initialize ε with very small positive value;
 - (2) Calculate global deviation σ ;
 - (3) Calculate local standard deviation σ_u ;
 - (4) **if** $\sigma_u \leq \sigma$ **then**
 - (5) Begin learning with loss function $\mathcal{L}$ according to Equation (17);
 - (6) **while** $e_{ui} > \varepsilon$ **do**
 - (7) $b_u \leftarrow b_u - \gamma_1 \left(\sum_{i \in I_u} e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot b_u \right)$;
 - (8) $b_i \leftarrow b_i - \gamma_1 \left(\sum_{u \in U_i} \left(\frac{\sigma_u}{\sigma} \right) \cdot e_{ui} + \lambda |U_i|^{-\frac{1}{2}} \cdot b_i \right)$;
 - (9) $\sigma_u \leftarrow \sigma_u - \gamma_1 \left(\sum_{i \in I_u} \left(\frac{b_i}{\sigma} \right) e_{ui} + \lambda |I_u|^{-\frac{1}{2}} \cdot \sigma_u \right)$;
 - (10) $p_u \leftarrow p_u - \gamma_2 \left(\sum_{i \in I_u} e_{u,i} q_i + \lambda_t \sum_{v \in T_u} e_{u,v} w_v + \left(\lambda |I_u|^{-\frac{1}{2}} + \lambda_t |T_u|^{-\frac{1}{2}} \right) p_u \right)$;
 - (11) $q_i \leftarrow q_i - \gamma_2 \left(\sum_{u \in U_i} e_{u,i} (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j + |T_u|^{-\frac{1}{2}} \sum_{v \in T_u} w_v) + \lambda |U_i|^{-\frac{1}{2}} q_i \right)$;
 - (12) $\forall j \in I_u, y_j \leftarrow y_j - \gamma_2 \left(\sum_{i \in I_u} e_{ui} |I_u|^{-\frac{1}{2}} q_i + \lambda |U_j|^{-\frac{1}{2}} y_j \right)$;
 - (13) $\forall v \in T_u, w_v \leftarrow w_v - \gamma_2 \left(\sum_{i \in I_u} e_{u,i} |T_u|^{-\frac{1}{2}} q_i + \lambda_t e_{u,v} p_u + \lambda |T_v^+|^{-\frac{1}{2}} w_v \right)$;
 - (14) **end while**
 - (15) Calculate $\hat{r}_{ui}$ according to Equation (16);
 - (16) **else**
 - (17) Begin learning with loss function according to algorithm in TrustSVD;
 - (18) Calculate $\hat{r}_{ui}$ according to Equation (15) of TrustSVD;
 - (19) **end if**
 - (20) **return** $\hat{r}_{ui}$
-

6. Experiments and Analysis

In this section, we performed a series of experiments to evaluate the efficiencies of the proposed SDP and its application instances (SDPSVD++ and SDPTrustSVD) on five real datasets, including Flixster, FilmTrust, MiniFilm, Movielens-10 M, and Movielens-latest-small (100K).

The evaluation performance of predictive accuracies was measured by the metrics mean absolute error (MAE), root mean square error (RMSE), and relative error (RE), respectively, where a lower value indicates better predictive accuracy. It was assumed $\tau = \{(u, i) | \exists r ((u, i, r) \in \mathbb{R})\}$, and N is the number of observed ratings. The evaluation matrices are as follows:

$$MAE = \frac{\sum_{(u,i) \in \tau} |r_{ui} - \hat{r}_{ui}|}{N}, \quad (18)$$

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in \tau} (r_{ui} - \hat{r}_{ui})^2}{N}}, \quad (19)$$

$$RE = \sqrt{\frac{\sum_{(u,i) \in \tau} (r_{ui} - \hat{r}_{ui})^2}{\sum_{(u,i) \in \tau} (r_{ui})^2}}. \quad (20)$$

We designed three experiments to evaluate the performance.

- Experiment (1): The first experiment evaluates the baseline estimation accuracy of the proposed SDP on the five datasets, compared with classical baseline estimation in (2) and the PBEModel [14].
- Experiment (2): The second experiment evaluates the improved accuracies of SDPSVD++ and proves the efficiency of SDP on improving existing CF recommendations.
- Experiment (3): The third experiment evaluates the performance of the proposed SDPTrustSVD based on the Flixster and FilmTrust datasets, which have trust ratings.

Each dataset in the experiments was separated into five parts. To perform a full evaluation, we used a cross-validation method during experiments in which four parts of a given dataset were used for training while the remaining part was used for prediction. As a result, each part was used for both training and prediction by cross validation. The final performance on a given dataset was expressed as the average of the cross-prediction results.

As mentioned above, our experiments were performed on five real datasets, and Table 3 shows their statistics.

Table 3. Details of the datasets for the experiments.

DataSets	#Users	#Items	#Ratings	#Density	Rating Scale	#Trust
Flixster	147,612	48,794	8,196,077	0.11%	[0.5,5]	11,794,648
FilmTrust	1508	2071	35,497	1.14%	[0.5,4]	1853
MiniFilm	55	334	1000	5.44%	[0.5,4]	0
ml-10 m	71,567	10,681	10,000,054	1.31%	[1,5]	0
ml-latest-small	700	10,000	100,000	1.43%	[1,5]	0

These datasets are all about movies. Flixster, Movielens (10 M), and Movielens (Latest Small) are generally large datasets, while FilmTrust and MiniFilm are relatively small. All of these could be found in related websites or www.librec.net. Flixster has 147 K users and 48 K movies with 409 K ratings and more than 11 M trust ratings with friendship links. Movielens (10 M) has more than 70 K users and 10 K movies with 10 M ratings. Movielens (Latest Small) is the latest small dataset of Movielens and has 700 users and 10 K movies with 100 K ratings. FilmTrust is relatively small and has 1508 users and 2071 items with 35,497 ratings and 1853 trust ratings. The smallest dataset is MiniFilm, which has only 55 users and 334 items with 1K ratings. Flixster and FilmTrust can be used to evaluate the performance of TrustSVD and SDPTrustSVD, which require latent factors from trust ratings.

6.1. Performance of Proposed SDP

The purpose of this experiment was to evaluate the baseline estimation accuracy of SDP on all users with full ratings using cross validation. We selected the classical baseline estimation (BE) described in (2) and the PBEModel [14] for baseline estimation performance comparison. We performed the experiment on the full ratings in each dataset. The evaluation results of MAEs, RMSEs, and REs are shown in Table 4 where PBE is short for PBEModel, and the visualized performance comparisons are shown in Figure 4.

Table 4. Baseline estimation performance of the proposed SDP.

Data Sets	Metrics	BE	PBE	Proposed SDP	Improved (vs. BE)
Flixster	RMSE	0.98813	0.95974	0.95890 *	2.96%
	RE	0.25145	0.24423	0.24402 *	2.95%
	MAE	0.73589	0.70248	0.69719 *	5.26%
FilmTrust	RMSE	0.84009	0.82623 *	0.83027 **	1.17%
	RE	0.26753	0.26312 *	0.26441 **	1.17%
	MAE	0.63667	0.62725 *	0.62806 **	1.35%
MiniFilm	RMSE	1.01431	1.00715	0.99605 *	1.80%
	RE	0.32314	0.32086	0.31733 *	1.80%
	MAE	0.78123	0.77251	0.76472 *	2.11%
ml_10m	RMSE	0.89921	0.89831	0.89162 *	0.84%
	RE	0.24329	0.24186	0.24039 *	1.19%
	MAE	0.68465	0.68451	0.67913 *	0.81%
ml_latest_small	RMSE	0.91483	0.89657	0.89607 *	2.05%
	RE	0.24923	0.24426	0.24412 *	2.05%
	MAE	0.70022	0.68840	0.68515 *	2.15%

Note: * is the best value, ** is the second.

The proposed SDP exhibits the best performances on Flixster, MiniFilm, Movielens (10 M), and Movielens (Latest Small), and second on FilmTrust. Compared with classical BE, the baseline predictive accuracies of the proposed SDP are superior. The MAEs of SDP are 5.26%, 1.35%, 2.11%, 0.81%, and 2.15% higher than classical BE on the datasets Flixster, FilmTrust, MiniFilm, Movielens (10 M), and Movielens (Latest Small), respectively. On the large-scale dataset Flixster, compared with BE, the RMSE and RE are also improved by 2.96% and 2.95%, respectively, by SDP. On another large-scale dataset, MovieLens (10 M), the improvements of RMSE, RE, and MAE are 0.84%, 1.19%, and 0.81%, respectively.

Compared with classical PBE, the predictive accuracies of SDP are also superior on Flixster, MiniFilm, Movielens (10 M), and Movielens (Latest Small), except for the small dataset FilmTrust. The results prove that the proposed SDP could effectively improve the baseline estimation accuracy, and all MAEs, RMSEs, and REs are superior to BE.

6.2. Performance of Proposed SDPSVD++

The purpose of this experiment was to evaluate the predictive accuracy of the proposed SDPSVD++ on all users with full ratings using cross validation. SDPSVD++ is an improved recommendation algorithm based on SVD++ [5] by the proposed baseline estimation model SDP. PBESVD++ [14] is also an improved SVD++ based on the baseline estimation model PBE. Therefore, we selected SVD++ and PBESVD++ for comparisons. We performed the experiment on the full ratings in four datasets by

cross validation, including Flixster, FilmTrust, MiniFilm, and Movielens (Latest Small), with 5 and 10 dimensions, respectively. The evaluation results of MAEs, RMSEs, and REs are shown in Table 5, and the visualized performance comparisons on each dataset are shown in Figure 5. The results show that the predictive accuracies (RMSEs, REs, and MAEs) of the proposed SDPSVD++ are superior to those of SVD++ and PBESVD++.

Figure 4. Performances (MAEs, RMSEs, and REs) of baseline estimation of different models: BE, PBE, and proposed SDP. The results show that the proposed SDP is superior to classical BE on all datasets with full ratings evaluation.

Table 5. Performance of SDPSVD++.

Dataset	Metrics	SVD++	PBESVD++	SDPSVD++	Improved (vs. SVD++)
Flixster (d = 5)	RMSE	0.98347	0.93271	0.91550 *	6.91%
	RE	0.24994	0.23587	0.23283 *	6.85%
	MAE	0.72168	0.67795	0.66620 *	7.69%
Flixster (d = 10)	RMSE	0.93490	0.93256	0.91498 *	2.13%
	RE	0.23759	0.23583	0.23269 *	2.06%
	MAE	0.68984	0.67808	0.66591 *	3.47%
FilmTrust (d = 5)	RMSE	0.81488	0.80823	0.79726 *	2.16%
	RE	0.25950	0.25739	0.25389 *	2.16%
	MAE	0.63339	0.62120	0.61539 *	2.84%

Table 5. Cont.

Dataset	Metrics	SVD++	PBESVD++	SDPSVD++	Improved (vs. SVD++)
FilmTrust (d = 10)	RMSE	0.81474	0.80754	0.79735 *	2.13%
	RE	0.25946	0.25717	0.25392 *	2.14%
	MAE	0.63330	0.61536 *	0.61550	2.81%
MiniFilm (d = 5)	RMSE	0.98790	0.98696	0.97232 *	1.58%
	RE	0.31090	0.31060	0.30598 *	1.58%
	MAE	0.76582	0.75788	0.75614 *	1.26%
MiniFilm (d = 10)	RMSE	0.98845	0.98749	0.97200 *	1.66%
	RE	0.31109	0.31078	0.30588 *	1.67%
	MAE	0.76924	0.75806	0.75568 *	1.76%
ml_latest_small (d = 5)	RMSE	0.87226	0.87089	0.86618 *	0.70%
	RE	0.23764	0.23727	0.23599 *	0.69%
	MAE	0.67298	0.67285	0.66847 *	0.67%
ml_latest_small (d = 10)	RMSE	0.87353	0.87223	0.86813 *	0.62%
	RE	0.23810	0.23775	0.23663 *	0.62%
	MAE	0.67307	0.67284	0.66871 *	0.65%

Note: * is the best value.

Compared with the original SVD++, using MAEs as examples, on each dataset with $d=5$, the predictive accuracies of SDPSVD++ are 7.69%, 2.84%, 1.26%, and 0.67% higher than those of SVD++. On each dataset with $d=10$, the improvements of SDPSVD++ are also distinct, with MAE improvements of 3.47%, 2.81%, 1.76%, and 0.65% compared with original SVD++. The greatest improvements appear in the large-scale dataset Flixster with $d = 5$, where the RMSE, RE, and MAE by SDPSVD++ are more than 6.8% higher than SVD++.

Compared with PBESVD++, the proposed SDPSVD++ also obtains higher predictive accuracies. Continuing to use MAEs for examples as described in Table 5, for each database with $d = 5$, the proposed SDPSVD++ are 1.73%, 0.94%, 1.49%, and 0.65% higher than those of PBESVD++. This proves that the proposed SDP can improve the recommendation accuracy of the existing SVD++ more efficiently than PBE.

Additionally, we observed that the predictive accuracies on full ratings of the small-scale dataset FilmTrust and MiniFilm are also improved greatly by SDPSVD++. On dataset FilmTrust, with $d = 5$ or $d = 10$, the RMSE, RE, and MAE are always more than 2.1% higher than SVD++. Compared with SVD++, the RMSE, RE, and MAE are 1.66%, 1.67%, and 1.76% higher, respectively, than those of the original SVD++ on MiniFilm with $d = 10$, proving that the SDP can be effectively combined with existing CF-based recommendation algorithms in different situations and obtain higher predictive accuracies.

6.3. Performance of Proposed SDPTrustSVD

The purpose of this experiment was to evaluate the predictive accuracy of the proposed SDPTrustSVD on all users with full ratings using cross validation. We selected the TrustSVD [11] for comparison. Because the two need to compute the trust information, we performed the experiment on the full ratings in Flixster and FilmTrust by cross validation, with $d = 5$ and $d = 10$, respectively. The evaluation results of MAEs, RMSEs, and REs are shown in Table 6, and the visualized performance comparisons on each dataset are shown in Figure 6.

Figure 5. Performances (MAEs, RMSEs, and REs) of predictive accuracies of different models: SVD++, PBESVD++, and proposed SDPSVD++. The results show that the proposed SDP can effectively improve the performance of SVD++.

Table 6. Performance of SDPTrustSVD.

Dataset	Metrics	TrustSVD	SDPTrustSVD	Improved (vs. TrustSVD)
Flixster (d = 5)	RMSE	0.92323	0.90564 *	1.91%
	RE	0.23451	0.23032 *	1.79%
	MAE	0.68654	0.66460 *	3.20%
Flixster (d = 10)	RMSE	0.92337	0.91548 *	0.85%
	RE	0.23455	0.23254 *	0.86%
	MAE	0.68669	0.67480 *	1.73%
FilmTrust (d = 5)	RMSE	0.81201	0.79262 *	2.39%
	RE	0.25856	0.25241 *	2.38%
	MAE	0.63938	0.61474 *	3.85%
FilmTrust (d = 10)	RMSE	0.81924	0.79148 *	3.39%
	RE	0.26086	0.25205 *	3.38%
	MAE	0.64495	0.61352 *	4.87%

Note: * is the best value.

Figure 6. Performances (MAEs, RMSEs, and REs) of the predictive accuracies of different models: TrustSVD, and proposed SDPTrustSVD. The results show that the proposed SDP can effectively improve the performance of TrustSVD.

The results show that the predictive accuracies of the proposed SDPTrustSVD are superior to the original TrustSVD. On full ratings of Flixster with $d = 5$, the improvements of RMSE, RE, and MAE by SDPTrustSVD are 1.91%, 1.79%, and 3.20% higher than the original Trust SVD. On FilmTrust with $d = 5$, the proposed SDP could also improve TrustSVD effectively, where the performances of RMSE, RE, and MAE are 2.39%, 2.38%, and 3.85% higher than TrustSVD. On both datasets with $d = 10$, the improvements are also distinct. The results prove that the proposed SDP can efficiently improve the existing TrustSVD.

7. Conclusions

In this paper, we investigated how to improve the baseline estimation accuracy. Observed from real datasets, we found that different users usually have different rating criteria, and the real rating range of users is usually a subset of the system's rating range. Further observations from ratings' global and local distributions showed that ratings generally follow the normal distribution and different personalized rating ranges result in different standard deviations. Moreover, we analyzed the ratings' distribution and found that the proportion of users' personalized local deviation σ_u to the system's global deviation σ can improve the baseline estimation accuracy and proposed an improved novel model named SDP, which can be conveniently applied to all existing latent factor-based CF recommendations that utilize classical baseline estimation, by adding a proportion $\left(\frac{\sigma_u}{\sigma}\right)$ to the global b_i . Two application instances of SDP, SDPSVD++ and SDPTrustSVD, were proposed in this paper to

illustrate how to apply SDP and how to improve the learning algorithms. The experiments showed that the proposed SDP is superior to classical baseline estimation on all datasets with full ratings evaluation by cross validation. The experiments also showed that the proposed instances SDPSVD++ and SDPTrustSVD have dramatical improvements on predictive accuracies compared to original algorithms. The results prove that SDP can not only improve baseline estimation accuracy effectively but also efficiently improve existing latent factor-based CF recommendations.

Author Contributions: Conceptualization, Z.T. and L.H.; methodology, Z.T.; formal analysis, L.H. and Q.C.; data curation, D.W.; validation, D.W. and Q.C.; writing—original draft preparation, Z.T. and L.H.; writing—review and editing, B.Z.; project administration, B.Z.; funding acquisition, Z.T.; supervision, Z.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grants No. 61772125 and No. 61402097; The National Key Research and Development Program of China under Grant No. 2019YFB1405803; CERNET Innovation Project under Grant No. NGII20190609.

Acknowledgments: All datasets supported by Open LibRec (<https://www.librec.net/datasets.html>). An earlier version of this paper was presented at ICME'19, Shanghai, China. Thank Fernando Ortega Requena for inviting us to submit this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jalili, M.; Ahmadian, S.; Izadi, M.; Moradi, P.; Salehi, M. Evaluating Collaborative Filtering Recommender Algorithms: A Survey. *IEEE Access* **2018**, *6*, 74003–74024. [CrossRef]
2. Tan, Z.; He, L. An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle. *IEEE Access* **2017**, *5*, 27211–27228. [CrossRef]
3. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
4. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. Major components of the gravity recommendation system. *ACM SIGKDD Explor. Newsl.* **2007**, *9*, 80–83. [CrossRef]
5. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. Available online: https://www.cs.rochester.edu/twiki/pub/Main/HarpSeminar/Factorization_Meets_the_Neighborhood-_a_Multifaceted_Collaborative_Filtering_Model.pdf (accessed on 10 May 2016).
6. Hernando, A.; Bobadilla, J.S.; Ortega, F.; Tejedor, J. Incorporating reliability measurements into the predictions of a recommender system. *Inf. Sci.* **2013**, *218*, 1–16. [CrossRef]
7. Moradi, P.; Ahmadian, S. A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Syst. Appl.* **2015**, *42*, 7386–7398. [CrossRef]
8. Koren, Y. Collaborative Filtering with Temporal Dynamics. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.379.1951&rep=rep1&type=pdf> (accessed on 10 May 2016).
9. Kumar, R.; Verma, B.K.; Rastogi, S.S. Social popularity based SVD++ recommender system. *Int. J. Comput. Appl.* **2014**, *87*, 33–37. [CrossRef]
10. Bao, Y.; Fang, H.; Zhang, J. TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation. Available online: <https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/viewFile/8273/8391> (accessed on 15 September 2015).
11. Guo, G.; Zhang, J.; Yorke-Smith, N. TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. Available online: <https://guoguibing.github.io/papers/guo2015trustsvd.pdf> (accessed on 20 June 2016).
12. Guo, G.; Zhang, J.; Yorke-Smith, N. A novel recommendation model regularized with user trust and item ratings. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1607–1620. [CrossRef]
13. Pan, W.; Yang, Q.; Duan, Y.; Ming, Z. Transfer learning for semisupervised collaborative recommendation. *ACM TIS* **2016**, *6*, 1–21. [CrossRef]
14. Tan, Z.; He, L.; Li, H.; Wang, X. Rating Personalization Improves Accuracy: A Proportion-Based Baseline Estimate Model for Collaborative Recommendation. In *Computing: Networking, Applications and Worksharing: 12th International Conference, CollaborateCom 2016, Beijing, China, November 10–11, 2016, Proceedings*; Springer: Cham, Switzerland, 2017.

15. Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R. A content-based recommender system for computer science publications. *Knowl. Based Syst.* **2018**, *157*, 1–9. [\[CrossRef\]](#)
16. Zhang, Y.; Koren, J. Efficient Bayesian hierarchical user modeling for recommendation system. Available online: <https://users.soe.ucsc.edu/~jyz/papers/c10-sigir07.pdf> (accessed on 10 May 2015).
17. Horváth, T. A model of user preference learning for content-based recommender systems. *Comput. Inform.* **2012**, *28*, 453–481.
18. Sharma, R.; Gopalani, D.; Meena, Y. Collaborative filtering-based recommender system: Approaches and research challenges. In Proceedings of the 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017.
19. Shih, Y.Y.; Liu, D.R. Product recommendation Approaches: Collaborative filtering via customer lifetime Value and customer demands. *Expert Syst. Appl.* **2008**, *35*, 350–360. [\[CrossRef\]](#)
20. Badaro, G.; Hajj, H.; El-Hajj, W.; Nachman, L. A hybrid approach with collaborative filtering for recommender systems. In Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013.
21. Parhi, P.; Pal, A.; Aggarwal, M. A survey of methods of collaborative filtering techniques. In Proceedings of the 2017 International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2017.
22. Cao, G.; Kuang, L. Identifying Core Users based on Trust Relationships and Interest Similarity in Recommender System. In Proceedings of the 2016 IEEE International Conference on Web Services (ICWS), San Francisco, CA, USA, 27 June–2 July 2016.
23. Bartolini, I.; Zhang, Z.; Papadias, D. Collaborative Filtering with Personalized Skylines. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 190–203. [\[CrossRef\]](#)
24. Li, W.; Xu, H.; Ji, M.; Xu, Z.; Fang, H. A hierarchy weighting similarity measure to improve user-based collaborative filtering algorithm. In Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October 2016.
25. Qian, X.; Feng, H.; Zhao, G.; Mei, T. Personalized Recommendation Combining User Interest and Social Circle. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1763–1777. [\[CrossRef\]](#)
26. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by latent semantic analysis. *J. Am. Soc. Inform. Sci.* **1990**, *41*, 391–407. [\[CrossRef\]](#)
27. Wu, D.; Luo, X.; Shang, M.; He, Y.; Wang, G.; Zhou, M. A Deep Latent Factor Model for High-Dimensional and Sparse Matrices in Recommender Systems. *IEEE Trans. Syst. Man Cybern.* **2019**. [\[CrossRef\]](#)
28. Li, S.; Kawale, J.; Fu, Y. Deep collaborative filtering via marginalized denoising auto-encoder. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne Australia, 19–23 October 2015.
29. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* **2019**, *52*, 5. [\[CrossRef\]](#)
30. Sofiane, B.B.; Bermak, A. Gaussian process for nonstationary time series prediction. *Comput. Stat. Data Anal.* **2004**, *47*, 705–712. [\[CrossRef\]](#)
31. Ko, J.; Fox, D. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Auton. Robot.* **2009**, *26*, 75–90. [\[CrossRef\]](#)
32. Tan, Z.; Wu, D.; He, L.; Chang, Q.; Zhang, B. SDP: An Improved Baseline Estimation Model Based on Standard Deviation Proportion. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019.