A Biased Proportional-Integral-Derivative-Incorporated Latent Factor Analysis Model

Jialu Sui and Jian Yin

School of Mechanical and Information Engineering, Shandong University, Weihai 264209, China; 20170800459@mail.sdu.edu.cn (* Correspondence: yinjian@sdu.edu.cn; Tel.: +086-18663130930

Abstract: Nowadays, as the number of items is increasing and the number of items that users have access to is limited, user-item preference matrices in recommendation systems are always sparse. This leads to a data sparsity problem. The latent factor analysis (LFA) model has been proposed as the solution to the data sparsity problem. As the basis of the LFA model, the singular value decomposition (SVD) model, especially the biased SVD model, has great recommendation effects in high-dimensional sparse (HiDs) matrices. However, it has the disadvantage of requiring several iterations before convergence. Besides, the model PID-incorporated SGD-based LFA (PSL) introduces the principle of discrete PID controller into the stochastic gradient descent (SGD), the learning algorithm of the SVD model. It could solve the problem of slow convergence speed, but its accuracy of recommendation needs to be improved. In order to make better solution, this paper fuses the PSL model with the biased SVD model, hoping to obtain better recommendation result by combining their advantages and reconciling their disadvantages. The experiments show that this biased PSL model performs better than the traditional matrix factorization algorithms on different sizes of datasets.

Keywords: latent factor analysis; stochastic gradient descent; PID controller; biased SVD model

1. Introduction

With the development of the network and the arrival of information times, people are exposed to more and more information. However, the number of items that users are interested in does not increase from before. It only accounts for a small proportion of the total items. Currently, it is difficult to satisfy the needs of most users by allowing them to go through the filter by themselves or by filtering only with item labels. In this case, it becomes essential to design a specific recommendation system to help users find the things they might be enthusiastic about. The recommendation system can extract the user’s preference information and recommend items that fit each user’s interest. The introduction of the recommendation system can significantly enhance the user’s experience with the software as they can effortlessly meet their own needs. Moreover, it has great commercial value in the fields of advertising promotion and commodity sales.

The main function of the recommendation system [1,2] is to predict user’s ratings through a series of calculations according to the existing scores and then fill in the user-item matrix with the predicted scores. Most of the data in the matrix are vacant because the amount of scoring data are much lesser than the number of users multiplied by the number of items. The recommendation system needs to fill the matrix to acquire a data-filled user-item matrix, which entails obtaining all users’ scores on all items.

Owing to the amount of information increases and the number of items each user has access to is limited, the user-item preference matrix is always described as the high-dimensional sparse (HiDS) matrix. For example, the MovieLens-10M dataset contains a total of 72,000 users and 10,000 movies. The user rating data shows that each user watched and rated only about 140 movies on average, with a density of only 1.31%. Compared with the total number of movies, the proportion of movies with user ratings is too low. In this
situation, cold start [3], reduced coverage, neighbor transitivity and a series of problems need to be solved. Data sparsity [4] brings tons of difficulties to the recommendation. In other words, the user provides little information for reference by the recommendation system, but the number of items that are needed to predict the scores is quite large. In previous studies, scholars have proposed many solutions to these problems, such as singular value decomposition (SVD) [5], principal component analysis (PCA) [6], content-boosted CF algorithm (CBCF) [7], tree augmented naïve bayes optimized by extended logistic regression (TAN-ELR) [8]. All of them have their pros and cons.

Aiming to devise better solution to the data sparsity problem, this paper describes a recommendation model based on the latent factor analysis (LFA) model, which is a kind of model-based recommendation techniques of the collaborative filtering (CF) recommendation system [9]. It uses the idea of the biased SVD model to improve the PID-incorporated SGD-based LFA (PSL) model which combines the advantages of both of them. This model fuses the prediction method of biased SVD with the instantaneous error correction method of PSL, which solves the shortcomings of slow convergence of the biased SVD model and the problem of low recommendation accuracy of the PSL model. When compared with other recommendation models, experiments prove that it has high computational efficiency, as well as high prediction accuracy on the HiDS matrix. The contributions of this paper are as follows:
1. It introduces a biased PSL model combining the biased SVD and the PSL model;
2. Experiments on three large datasets demonstrate that the biased PSL model can achieve highly competitive prediction accuracy for the missing data of an HiDS matrix compared to the other models.

The rest of this paper is organized as follows: Section 2 shows the related work about the research; Section 3 gives the preliminaries knowledge for a detailed introduction to the basis of the algorithm. Section 4 describes the specific implementation process of the algorithm. Section 5 reports the recommendation results of the algorithm and compares it with previous recommendation methods; Section 6 evaluates the performance of each method; Finally, Section 7 summarizes the whole paper and puts forward the direction of future research.

2. Related Work

The recommendation system is generally divided into 3 main categories: (1) Content-based recommendation system [10,11]. It recommends the items that are similar to previous favorite ones to users based on item content and item label; (2) CF Recommendation System [12–14]. It gives priority to recommend the items that are favored by people who have similar preferences to them and the recommendation is based on the relevance of users and items; (3) Hybrid Recommendation System [15,16]. It combines multiple techniques to make the final decision.

As for the CF algorithms, it can be divided into 3 techniques: (1) Memory-Based CF Techniques [17]. For these techniques, every user is part of a group of people with similar interests. A new user’s preference could be obtained by identifying his so-called neighbors; (2) Model-Based CF Techniques [18]. In order to solve the limitations of memory-based CF algorithms, it applies some models to memory-based CF algorithms, such as machine learning and data mining algorithms. It can resolve the limitations of memory-based CF algorithms; (3) Hybrid CF Techniques [19,20]. As each recommendation technology has its limitations, it combines multiple recommendation techniques to make the final recommendation.

As one of the model-based CF recommendation technique and the basis of the model introduced in this paper, the LFA model [21–25] decomposes the user-item matrix into $P_{u\times k}$ and $Q_{i\times k}$ by the SVD, where $u$ and $i$ are the number of the users and items, respectively, and $k$ is the number of latent factors. Then, it obtains the final prediction scoring matrix by multiplying the two matrices. The value of $k$ is self-set, which refers to the number of latent factors. The item matrix can be understood as the degree to which each item has
this series of attributes, while the user matrix can be considered as the degree to which each user likes this series of attributes. However, to say “each latent factor represents an attribute” is just for the convenience of understanding. In fact, each latent factor has no definite meaning, which makes the latent factor model more flexible and not limited to the label attribute of the item.

According to the previous studies [26,27], the SVD model, especially the biased SVD model, is proposed to solve a series of problems of data sparsity since it has a good performance on the HiDS matrix. The biased SVD model takes users’ rating habits and the quality of items into account to give a more targeted prediction score in accordance with the characteristics of users and items. In the Netflix Prize recommendation game, Yehuda Koren reduced the rating error by 32% when adding the bias portion alone, and by 42% when adding the personalization portion. Thus, by adding personalization, it can only reduce the error by 10%. This illustrates the importance of the bias which has a greater effect in improving accuracy. However, this model has a serious drawback, it requires multiple iterations to converge on large datasets.

In addition to the biased SVD model, according to [28,29], based on the SVD model, the LFA model incorporating the principle of discrete PID controller named PSL is also raised to solve data sparsity because it has the advantage of rapid convergence. The traditional SVD can be regarded as a discrete PID controller with $K_I = K_D = 0$. Therefore, this method adds these two parameters to improve the performance of the recommendation model. However, since it assumes that everyone has the same rating standard and all items have the same quality, its recommendation accuracy has to be improved.

Since both of them have advantages and disadvantage, this paper introduces the principle of the biased SVD model to the PSL model to improve the recommendation effect.

3. Preliminaries

The biased PSL model proposed in this paper incorporates the benefits of biased SVD and PSL models while also addressing their drawbacks. The following section introduces the SVD model and PSL model in detail.

3.1. SVD

3.1.1. Conventional Matrix Decomposition SVD Model

The SVD in the recommendation system is slightly different from the mathematical singular value decomposition [30]. In linear algebra, SVD decomposes a matrix into three matrices. As shown in (1), $W$ is the original matrix, $U$ is the left singular matrix, $V$ is the right singular matrix, and $\Sigma$ is the singular value diagonal matrix.

$$ W = U \Sigma V. $$

However, in the recommendation algorithm, SVD decomposes a matrix into two matrices $P_{uk}$ and $Q_{ik}$, the user latent factor matrix and the item latent factor matrix, respectively. Then it gains a matrix full of data which is a predicted matrix filled with user ratings for items through the product of two matrices like (2) and (3). This SVD method is the basis of the LFA.

$$ R_{ui} = PQ^T. $$

$$ \hat{r}_{ui} = \sum_{k=1}^{K} p_{u,k}q_{i,k}. $$

The recommendation system aims to minimize the error between the real score and the predicted score, thus the objective function of the traditional SVD model is (4). To avoid
the problem of gradient disappearance of \( P \) and \( Q \) in the process of gradient descent, the regularization term should be added. \( \lambda \) expresses the regularization parameter.

\[
\min_{Q^*,P^*} \sum_{(u,i) \in R} (r_{ui} - \sum_{k=1}^{K} p_{uk}q_{ik})^2 + \lambda(\|Q_i\|^2 + \|P_u\|^2). \tag{4}
\]

The iterative formulas (5) for \( P \) and \( Q \) are obtained by taking the derivative of the objective function.

\[
\begin{align*}
\frac{\partial \mathcal{C}}{\partial p_{uk}} &= -2(r_{ui} - \sum_{k=1}^{K} p_{uk}q_{ik})Q_{ki} + 2\lambda p_{uk}, \\
\frac{\partial \mathcal{C}}{\partial q_{ki}} &= -2(r_{ui} - \sum_{k=1}^{K} p_{uk}q_{ik})P_{uk} + 2\lambda Q_{ki}.
\end{align*}
\tag{5}
\]

Then the stochastic gradient descent (SGD) method (6) is used to acquire the final solution where \( \alpha \) represents the learning rate [31].

\[
\begin{align*}
p_{uk}^{t+1} &= p_{uk}^t + \alpha ((r_{ui} - \sum_{k=1}^{K} p_{uk}q_{ik})Q_{ki} - \lambda p_{uk}^t), \\
q_{ki}^{t+1} &= q_{ki}^t + \alpha ((r_{ui} - \sum_{k=1}^{K} p_{uk}q_{ik})P_{uk} - \lambda Q_{ki}^t).
\end{align*}
\tag{6}
\]

### 3.1.2. Biased SVD Model

Obviously, there are many subjective factors in the scoring process. Different people have different scoring habits. For example, some people like to give high scores, while others always give low scores. In addition, the quality of items varies from each other. Naturally, high-quality items always get higher scores than low-quality items. Under this circumstance, it is necessary to introduce bias into the SVD model. In the biased SVD model, the characteristics of the user and item are considered by adding an item bias and a user bias. Compared with the traditional SVD algorithm, the performance of the biased SVD algorithm is better, because it takes individuation into account. As mentioned above, the prediction score formula of the biased SVD model is shown in (7) where \( b_i \) and \( b_u \) indicate the user bias and item bias, representing the eigenvalues of users and items, respectively, and \( \mu \) expresses the average score of all users.

\[
p_{ui} = \sum_{k=1}^{K} p_{uk}q_{ik} + \mu + b_i + b_u. \tag{7}
\]

After gaining the predicted score, the objective function can be calculated as follows (8). Since \( b_i \) and \( b_u \) are also variables and need to be updated during iteration, the regularization terms of \( b_i \) and \( b_u \) need to be added as well.

\[
\min_{Q^*,P^*} \sum_{(u,i) \in R} (r_{ui} - \sum_{k=1}^{K} p_{uk}q_{ik} - \mu - b_i - b_u)^2 + \lambda(\|Q_i\|^2 + \|P_u\|^2 + \|b_i\|^2 + \|b_u\|^2). \tag{8}
\]

In the biased SVD model, since \( b_i \) and \( b_u \) have no product relationship with \( P \) and \( Q \), the iterative method of \( P \) and \( Q \) is the same as that of the traditional SVD method. By calculating the derivative of the objective function, the iterative formulas of \( b_i \) and \( b_u \) are obtained as follows (9), where \( \alpha \) indicates the learning rate.

\[
\begin{align*}
b_{ui}^{t+1} &= b_{ui}^t + \alpha((r_{ui} - p_{ui} - \lambda b_{ui}^t), \\
b_{li}^{t+1} &= b_{li}^t + \alpha((r_{ui} - p_{ui} - \lambda b_{li}^t).
\end{align*}
\tag{9}
3.2. **PSL Model**

As for the PSL model, it implements the principle of the PID controller into the LFA model which can improve the convergence speed. As such, before describing the PSL model, the principle of the PID controller is introduced at first.

### 3.2.1. PID Controller

The principle of PID [32,33] is to calculate instantaneous error, i.e., the difference between the true value and the predicted value, and then correct this error according to the proportion (P), integral (I), and derivative (D). Since the update point in the HiDS matrix is discrete, it is compatible with the discrete PID controller. The schematic diagram of the discrete PID controller is shown in Figure 1. The TV and PV in the figure represent true and predicted values, respectively.

![Flowchart of PID controller.](image)

As shown in Figure 1, firstly, it calculates the error between TV and PV, i.e., the instantaneous error, and then inputs it into the three modules, P, I, and D, to accomplish the following calculation (10). Through this formula we can attain the adjusted error, and finally use it to update the predicted value instead of using the instantaneous error.

\[
\tilde{E}_t = K_P E_t + K_I \sum_{n=0}^{t} E_n + K_D (E_t - E_{t-1}).
\] (10)

The control coefficients for the proportional, integral, and derivative terms are \(K_P\), \(K_I\), and \(K_D\), respectively, and \(E_t\) is the instantaneous error in the \(t\)th update point. The PV is reconstructed and returned to the controller based on the adjusted error. This mechanism continues until the termination condition, i.e., the LFA converges, is met.

### 3.2.2. PSL Model

To converge, the SGD-based LFA model requires several iterations, and the overall cost of time can be considerable. As a result, the convergence process must be accelerated. It can be shown that at each update point of the LFA model based on SGD, the instantaneous error between the actual value \(r_{ui}\) and the predicted value \(\tilde{\tau}_{ui}\) will be measured and transmitted to the algorithm. Therefore, this procedure can be considered as a generalized discrete PID controller, with \(K_I = K_D = 0\) omitting the integral and derivative terms. From this
standpoint, applying these two terms to the model will help it perform better and accelerate the convergence process. Based on the above assumptions, the PSL model is introduced. The main idea of the PSL model is to reconstruct the instantaneous error according to the PID principle, and then bring the reconstructed error into the SGD algorithm to accelerate the convergence of the LFA model. In the PSL model, according to the concept of PID controller, the integral term and the derivative term are used to extend the calculation of error, as seen in the formulas (11) and (12).

\[ \tau_{u,i} = r_{u,i} - \sum_{k=1}^{K} p_{u,k} q_{i,k}. \]  
\[ \tilde{\tau}_{u,i} = K_P \tau_{u,i}^{t} + K_I \sum_{n=0}^{t} \tau_{u,i}^{n} + K_D (\tau_{u,i}^{t} - \tau_{u,i}^{t-1}). \]  

- \( \sum_{n=0}^{t} \tau_{u,i}^{n} \) shows the sum of historical \( \tau_{u,i} \)
- \( \tau_{u,i}^{t} - \tau_{u,i}^{t-1} \) shows the discrepancy between this time’s error and last time’s error

The objective function (13) of the PSL model is deduced according to the above formulas.

\[ \min_{Q^*, P^*} \sum_{(u,i) \in R} \tilde{\tau}_{u,i}^2 + \lambda (\|Q_i\| + ||P_u||^2). \]  

By taking the derivative of the objective function, the following iterative formula (14) for \( P \) and \( Q \) is obtained.

\[
\begin{cases}
    P_{u,k}^{t+1} = P_{u,k}^{t+1} + \alpha (\tilde{\tau}_{u,i} Q_{ki} - \lambda P_{u,k}^t), \\
    Q_{ki}^{t+1} = Q_{ki}^{t+1} + \alpha (\tilde{\tau}_{u,i} P_{u,k} - \lambda Q_{ki}^t).
\end{cases}
\]  

4. Materials and Methods

This paper proposes a biased PSL model which combines the biased SVD model with the PSL model. It solves the problem that the PSL model does not consider individualization and the biased SVD model requires several iterations to convergence. The input of the model is the user-item matrix, \( K_P, K_I, K_D \) of the discrete PID controller, average score of all users, the number of users \( U \) and items \( I \). The output is the user latent factor matrix \( P \) and item latent factor matrix \( Q \). The final prediction scoring matrix is obtained by multiplying these two matrices.

4.1. Algorithm Description

According to [26–29], it can be concluded that both the PSL model and the biased SVD model help improve the effectiveness of the recommendation system. Therefore, this paper presents a new model combining the PSL model with the biased SVD model to test whether it can provide better recommendation result. In this method, the instantaneous error, i.e., the difference between the true value and the predicted value, is derived from the biased SVD formula, and the error is then fed into the PSL model for an iterative solution.

The calculation formula of the predicted value is the same as that of the biased SVD model (7). Then the instantaneous error value is brought into the PID controller to gain the adjusted error for iteration to obtain the final predicted value. In this way, the advantages of rapid convergence and fewer iterations of the PSL model can be combined with the advantage of high accuracy of the biased SVD model. Substituting the above predicted formula into the PSL model can obtain the formula (15).

\[ \tau_{u,i} = r_{u,i} - \sum_{k=1}^{K} p_{u,k} q_{i,k}, b_{u} - b_{i} \]
Algorithm 1 Bias-PSLSDV.

Require: $U, I, R, K, \lambda, \alpha, K_p, K_l, K_D, S, \mu$
Ensure: $P, Q$

init $P_{[0,0,K],Q_{[0,0,K]}$ with random numbers in $[-0.01, 0.01]$
init $b_u, b_i$ with 0
init $\Psi, Y$ with size $|R|$
for each $r_{u,i}$ in $R$ do
  init $\Psi_{u,i} = 0, Y_{u,i} = 0$
end for
while not converge and $n \leq S$ do
  for $r_{u,i}$ in $R$ do
    fetch $\Psi_{u,i}$ from $\Psi$ and $Y_{u,i}$ from $Y$
    $r_{u,i} = \sum_{k=1}^{K} p_{u,k} q_{i,k} + \mu + b_i + b_u$
    $\tau_{u,i} = r_{u,i} - \hat{r}_{u,i}$
    $\Psi_{u,i} = \Psi_{u,i} + \tau_{u,i}$
    $\bar{\tau} = K_p * \tau_{u,i} + K_l * \Psi_{u,i} + K_D * (\tau_{u,i} * Y_{u,i})$
    $Y_{u,i} = \tau_{u,i}$
    $b_u = b_u + \alpha(\bar{\tau} - \lambda b_u)$
    $b_i = b_i + \alpha(\bar{\tau} - \lambda b_i)$
  for $k = 1$ to $K$ do
    $p_{u,k} = p_{u,k} + \alpha(\bar{\tau} q_{i,k} - \lambda p_{u,k})$
    $q_{i,k} = q_{i,k} + \alpha(\bar{\tau} p_{u,k} - \lambda q_{i,k})$
  end for
end for
$n = n + 1$
end while

4.2. Algorithm Analysis

Since this algorithm is mainly based on the framework of the PSL model, the computational cost of the algorithm is quite similar to the PSL algorithm. In terms of storage complexity, it has two more sequences ($b_u$ and $b_i$) than that of the PSL algorithm. Apart from these, there is no other change. This storage method is suitable for practical development, as it makes the algorithm formula more concise and comprehensible. The storage complexity is described as $S_{Bias-PSL}$ in (18). $|U|$ and $|I|$ mean the number of users and the number of items, respectively, $K$ expresses the number of latent factors, and $|R|$ expresses the amount of data.

$$S_{Bias-PSL} = (|U| + |I|)(K + 1) + 2|R|$$

(18)
The computational cost corresponding to the storage complexity is described as $T_{Bias-PSL}$ in (19). In this formula, step is the number of iterations in the recommendation algorithm.

$$T_{Bias-PSL} = step \times |R| \times K$$  

(19)

It can be seen that the time complexity of the algorithm is mainly determined by the number of iterations, the amount of data and the number of latent factors.

5. Experiment Results

5.1. General Settings

The main function of the LFA model is to decompose the scoring matrix of missing data into two latent factor matrices $P$ and $Q$, the user latent factor matrix and item latent factor matrix, and finally calculate the product of the two latent factor matrices $PQ^T$ to obtain a complete scoring matrix. After gaining the complete scoring matrix, we remove the items users have already rated, and then rank the remaining items according to their predicted scores. Naturally, the items with high scores will be recommended first. As a result, accurate item ratings are the key to the recommendation system. The ultimate purpose of the LFA model is to gain missing data in the user-item matrix, so the main method to judge the recommendation effect of a recommendation model is to evaluate the difference between its predicted value and real value. At present, the methods commonly used are root mean square error (RMSE) [34] and mean absolute error (MAE) [35]. In Formula (20), $r_{ui}$ and $\hat{r}_{ui}$ denote the real score and predicted score, respectively, and $|R|$ denotes the amount of data.

$$\left\{ \begin{array}{l} RMSE = \sqrt{\frac{\sum_{(u,i)\in R}(r_{ui} - \hat{r}_{ui})^2}{|R|}}, \\ MAE = \frac{\sum_{(u,i)\in R}|r_{ui} - \hat{r}_{ui}|_{abs}}{|R|}. \end{array} \right.$$  

(20)

According to the formulas of the two methods, it can be concluded that the smaller the RMSE and MAE value of this model, the better its recommendation effect.

5.2. Dataset

The experiments use a dataset from the Movielens [36] series, which was collected from the Movielens system maintained by the GroupLens research team, with a score on the scale of 1 to 5 (1,2,3,4,5). To explore the recommendation effect of the biased PSL model on datasets of different sizes and densities, this experiment selects 3 different sizes of subsets of the dataset. Their details are provided in Table 1.

| Dataset | Users | Item | Ratings | Density |
|---------|-------|------|---------|---------|
| 100 K   | 943   | 1682 | 100,000 | 6.30%   |
| 1 M     | 6040  | 3952 | 1,000,209 | 4.19% |
| 10 M    | 71,567 | 10,681 | 10,000,054 | 1.31% |

5.3. Model Comparison Test

In this paper, the biased PSL model is compared with the traditional SVD model, the biased SVD model and the PSL model.

The learning rate of the algorithm is unified as 0.01, the regularization parameter is 0.05, $K_p$ is 1.44, $K_I$ is 0.002, and $K_D$ is 0.001. The emphasis is to compare the recommendation effects of different algorithms on different sizes of datasets under the condition of the different number of latent factors (50,100).

All models are accomplished based on surprise [37] (surprise is a Python scikit for building and analyzing recommender systems that deal with explicit rating data.) The models involved in the experiments are as follows:
• SVD model (the traditional SVD model): It decomposes the user-item matrix directly;
• Biased SVD model: It introduces the user and item bias into the SVD model, taking the individuation into account;
• PSL model: It introduces the principle of the discrete PID controller into the SGD and applies the parameter of PID to the traditional SVD model;
• Biased PSL model: It integrates the biased SVD model and the PSL model.

5.4. Results

The experiments adopt the 80–20% train-test settings and applies five-fold cross-validations to obtain objective results. The average training processes of compared models are shown in Figures 2–7.

Figure 2. Training process of compared models on MovieLen-100 K with 50 latent factors.

Figure 3. Training process of compared models on MovieLen-100 K with 100 latent factors.

Figure 4. Training process of compared models on MovieLen-1 M with 50 latent factors.
The performance of the four models on the datasets is shown below in Tables 2–4.

### Table 2. The prediction result on MovieLen-100 K.

| Method   | $K = 50$ | $K = 100$ |
|----------|----------|-----------|
|          | RMSE     | MAE       | RMSE     | MAE       |
| SVD      | 0.9368   | 0.7386    | 0.9340   | 0.7365    |
| BiasSVD  | 0.9272   | 0.7298    | 0.9198   | 0.7215    |
| PSL      | 0.9299   | 0.7307    | 0.9189   | 0.7214    |
| BiasPSL  | 0.9157   | 0.7186    | 0.9035   | 0.7065    |
Table 3. The prediction result on MovieLen-1 M.

| Method   | K = 50 | K = 100 |
|----------|--------|---------|
|          | RMSE   | MAE     | RMSE   | MAE     |
| SVD      | 0.8625 | 0.6817  | 0.8603 | 0.6805  |
| BiasSVD  | 0.8566 | 0.6733  | 0.8540 | 0.6715  |
| PSL      | 0.8635 | 0.6796  | 0.8589 | 0.6758  |
| BiasPSL  | 0.8506 | 0.6666  | 0.8459 | 0.6627  |

Table 4. The prediction result on MovieLen-10 M.

| Method   | K = 50 | K = 100 |
|----------|--------|---------|
|          | RMSE   | MAE     | RMSE   | MAE     |
| SVD      | 0.7973 | 0.6174  | 0.7943 | 0.6152  |
| BiasSVD  | 0.7916 | 0.6089  | 0.7906 | 0.6084  |
| PSL      | 0.7993 | 0.6134  | 0.7957 | 0.6108  |
| BiasPSL  | 0.7885 | 0.6040  | 0.7862 | 0.6024  |

6. Discussion

According to the Figures 2–7, the RMSE and MAE initially decrease as the number of iterations increases. However, once over-fitting occurs, the RMSE and MAE will rise as the number of iterations rises. In this experiment, the lowest point of the RMSE and MAE before over-fitting is chosen for comparison. From these results, we have following findings:

1. By comparing the 4 models in Figures 2–7, it can be seen that the biased PSL model requires fewer iterations to achieve the best recommendation result;
2. By comparing the 4 models in Figures 2–7, it can be seen that the final RMSE and MAE of the biased PSL model are smaller, which means it has better recommendation effect;
3. By comparing Figures 2, 4 and 6, as well as Figures 3, 5 and 7 we can conclude that the size of the dataset has little impact on the number of iterations needed to achieve the best recommendation effect of the biased PSL model.

The biased PSL model proposed in this paper can combine the advantages of the biased SVD model and the PSL model.

According to Tables 2–4, the biased PSL model outperforms the SVD model, the biased SVD model, and the PSL model in terms of the recommendation result in each dataset with different number of latent factors. Among the three datasets, the smallest dataset (100 K) can best reflect the advantages of the biased PSL model. As shown in Table 2, the RMSE of the biased PSL model decreased by 0.0227, about 1%, compared with the traditional SVD model, the model with the worst recommendation effect. Even if the difference is not extraordinarily obvious, its result is still superior to the biased SVD model and PSL model. From the data in Tables 2–4, it can be proven that the combination of the biased SVD model and the PSL model can result in improved performance.

7. Conclusions

This paper combines the prediction formula of the biased SVD model with the PSL model, designing a biased PSL model which achieves significantly higher computational efficiency, as well as highly competitive prediction accuracy. This method fuses the prediction method of biased SVD with the instantaneous error correction method of PSL. The biased SVD formula can take into account the scoring habits of different people and the different qualities of items, i.e., taking into account the individuation, to give the most appropriate score for users and items. The PSL model integrates the principle of the discrete PID controller into the SGD-based LFA model, to corrects the errors of the real and predicted scores according to the proportion (P), integral (I), and derivative (D). The biased PSL model blends the advantages of the two models to obtain a better recommendation effect. In the experiment, we use 3 datasets with different sizes and different densities to
test the recommendation performance of the biased PSL model and other recommendation models (SVD model, biased SVD model, and PSL model). The prediction effect is evaluated in detail by measuring the results of RMSE and MAE. Eventually, the experimental result delivers that the biased PSL model performs better than the SVD and PSL models on the three HiDS matrices.

Regarding future work, we hope to determine the best parameters appropriate for specific dataset. This model is mainly based on the PSL model, and the principle of the PSL model is discrete PID controller. Thus, its prediction results are greatly affected by the parameters of the PID controller. Owing to the wide range of $K_P$, $K_I$, and $K_D$, it is difficult to ensure whether $K_P$, $K_I$, and $K_D$ are most suitable parameters for this dataset. If a system can achieve the best parameters appropriate for the specific dataset, the best recommendation effect could be gained. Furthermore, despite having fewer iterations, the PSL algorithm takes longer to complete each iteration than the SVD algorithm due to the difficulty of vector changes in each iteration. Its capacity to run on a huge dataset is restricted, as it takes longer to complete a single training session. Since new users and items are be introduced regularly into a recommendation system, the lengthy preparation period is not conducive to the implementation of the model. It is difficult to label a recommendation system as outstanding if it cannot measure easily and make initial suggestions as new users and items are being introduced. Besides, we only attempted to combine the PSL model and the biased SVD model in this paper in the hope of a better recommendation effect. Now no attempt has been made to combine the PSL model with other recommendation models. In the future, we can try integrating PSL model with other recommendation models and test their recommendation performance.

Author Contributions: Conceptualization, J.S.; methodology, J.S.; software, J.S.; validation, J.S.; formal analysis, J.S.; writing—original draft preparation, J.S.; writing—review and editing, J.Y.; supervision, J.Y.; 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 Grant 61971268.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available online at https://github.com/littlebeen/Bias-PSL (accessed on 15 June 2021). And the datasets are available online at https://grouplens.org/datasets/movielens/ (accessed on 15 June 2021).

Acknowledgments: We thank the National Natural Science Foundation of China for funding our work, grant number 61971268.

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

Abbreviations
The following abbreviations are used in this manuscript:

| Abbreviation | Description |
|--------------|-------------|
| LFA          | Latent Factor Analysis |
| SVD          | Singular Value Decomposition |
| PSL          | PID-incorporated SGD-based LFA |
| SGD          | Stochastic Gradient Descent |
| CF           | Collaborative Filtering |
| PCA          | Principal Component Analysis |
| CBCF         | Content-Boosted CF Algorithm |
| TAN-ELR      | Tree Augmented Naïve Bayes optimized by Extended Logistic Regression |
| PID          | Proportion Integral Derivative |
| RMSE         | Root Mean Square Error |
| MAE          | Mean Absolute Error |
References

1. Abdollahpouri, H.; Malthouse, E.C.; Konstan, J.A.; Mobasher, B.; Gilbert, J. Toward the Next Generation of News Recommender Systems. arXiv 2021, arXiv:abs/2103.06909.

2. Li, Z. Towards the next generation of multi-criteria recommender systems. In Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, 2–7 October 2018; Pera, S., Ekstrand, M.D., Amatriain, X., O’Donovan, J., Eds.; ACM: New York, NY, USA, 2018; pp. 553–557. [CrossRef]

3. Shao, Y.; Xie, Y. Research on Cold-Start Problem of Collaborative Filtering Algorithm. In Proceedings of the 3rd International Conference on Big Data Research, ICBDR 2019, Cergy-Pontoise, France, 20–22 November 2019; ACM: New York, NY, USA, 2019; pp. 67–71. [CrossRef]

4. Yin, H.; Wang, Q.; Zheng, K.; Li, Z.; Zhou, X. Overcoming Data Sparsity in Group Recommendation. arXiv 2020, arXiv:abs/2010.00813.

5. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. Scalable Collaborative Filtering Approaches for Large Recommender Systems. J. Mach. Learn. Res. 2009, 10, 623–656.

6. Yazdani, S.; Shanbehzadeh, J.; Shalmani, M.T.M. RPCA: A Novel Preprocessing Method for PCA. Adv. Artif. Intell. 2012, 2012, 484595:1–484595:7. [CrossRef]

7. Lin, H.; Yang, X.; Wang, W. A Content-Boosted Collaborative Filtering Algorithm for Personalized Training in Interpretation of Radiological Imaging. J. Digit. Imaging 2014, 27, 449–456. [CrossRef] [PubMed]

8. Greiner, R.; Su, X.; Shen, B.; Zhou, W. Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. Mach. Learn. 2005, 59, 297–322. [CrossRef]

9. Su, X.; Khoshgoftaar, T.M. A Survey of Collaborative Filtering Techniques. Adv. Artif. Intell. 2009, 2009, 421425:1–421425:19. [CrossRef]

10. Balabanović, M.; Shoham, Y. Fab: Content-Based, Collaborative Recommendation. Commun. ACM 1997, 40, 66–72. [CrossRef]

11. Liu, H.; Kong, X.; Bai, X.; Wang, W.; Bekele, T.M.; Xia, F. Context-Based Collaborative Filtering for Citation Recommendation. IEEE Access 2015, 3, 1695–1703. [CrossRef]

12. Gogna, A.; Majumdar, A. A Comprehensive Recommender System Model: Improving Accuracy for Both Warm and Cold Start Users. IEEE Access 2015, 3, 2803–2813. [CrossRef]

13. Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, 1–5 May 2001; Shen, V.Y., Saito, N., Lyu, M.R., Zurko, M.E., Eds.; ACM: New York, NY, USA, 2001, pp. 285–295. [CrossRef]

14. Wu, Y.; Wei, J.; Yin, J.; Liu, X.; Zhang, J. Deep Collaborative Filtering Based on Outer Product. IEEE Access 2020, 8, 85567–85574. [CrossRef]

15. Nguyen, S.; Kwak, H.; Lee, S.; Gim, G.Y. Featured Hybrid Recommendation System Using Stochastic Gradient Descent. Int. J. Netw. Distrib. Comput. 2021, 9, 25–32. [CrossRef]

16. Cai, X.; Hu, Z.; Zhao, P.; Zhang, W.; Chen, J. A hybrid recommendation system with many-objective evolutionary algorithm. Expert Syst. Appl. 2020, 159, 113648. [CrossRef]

17. Petrusel, M. A Comparative Analysis of Similarity Measures in Memory-Based Collaborative Filtering. In Proceedings of the Tenth International Conference on Artificial Intelligence and Soft Computing—19th International Conference, ICAISC 2020, Zakopane, Poland, 12–14 October 2020; Barták, R., Brawner, K.W., Eds.; AAAI Press: Palo Alto, CA, USA 2019; pp. 419–424. [CrossRef]

18. Yang, F.; Liu, F.; Liu, S. Collaborative Filtering Based on a Variational Gaussian Mixture Model. Future Inter. 2021, 13, 37. [CrossRef]

19. Yue, W.; Wang, Z.; Tian, B.; Pook, M.; Liu, X. A Hybrid Model- and Memory-Based Collaborative Filtering Algorithm for Baseline Data Prediction of Friedreich’s Ataxia Patients. IEEE Trans. Ind. Informat. 2021, 17, 1428–1437. [CrossRef]

20. Wang, X.; Dai, Z.; Li, H.; Yang, J. Research on Hybrid Collaborative Filtering Recommendation Algorithm Based on the Time Effect and Sentiment Analysis. Complex 2021, 2021, 6635202:1–6635202:11. [CrossRef]

21. van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep, End-to-End, Text-to-Speech with a Phonetically Encoded Neural Simulation. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–8 December 2013; Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q., Eds.; 2013; pp. 2643–2651.

22. Luo, X.; Zhou, M.; Li, S.; You, Z.; Xia, Y.; Zhu, Q. A Nonnegative Latent Factor Model for Large-Scale Sparse Matrices in Recommender Systems via Alternating Direction Method. IEEE Trans. Neural. Netw. Learn. Syst. 2016, 27, 579–592. [CrossRef]

23. Costa, F.S.D.; Dolog, P. Convolutional Adversarial Latent Factor Model for Recommender System. In Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, FL, USA, 19–22 May 2019; Barták, R., Brawner, K.W., Eds.; AAAI Press: Palo Alto, CA, USA 2019; pp. 419–424. [CrossRef]

24. Mongia, A.; Jham, N.; Chouzenoux, É.; Majumdar, A. Deep latent factor model for collaborative filtering. Signal Process. 2020, 169, 107366. [CrossRef]

25. Shang, M.; Luo, X.; Liu, Z.; Chen, J.; Yuan, Y.; Zhou, M. Randomized latent factor model for high-dimensional and sparse matrices from industrial applications. IEEE CAA J. Autom. Sinica 2019, 6, 131–141. [CrossRef]

26. Luo, X.; Zhou, M.; Shang, M.; Li, S.; Xia, Y. A Novel Approach to Extracting Non-Negative Latent Factors From Non-Negative Big Sparse Matrices. IEEE Access 2016, 4, 2649–2655. [CrossRef]
27. Gogna, A.; Majumdar, A. SVD free matrix completion with online bias correction for Recommender systems. In Proceedings of the 2015 Eighth International Conference on Advances in Pattern Recognition (ICAIPR), Kolkata, India, 4–7 January 2015; pp. 1–5. [CrossRef]

28. Li, J.; Yuan, Y.; Ruan, T.; Chen, J.; Luo, X. A proportional-integral-derivative-incorporated stochastic gradient descent-based latent factor analysis model. *Neurocomputing* **2021**, *427*, 29–39. [CrossRef]

29. Li, J.; Yuan, Y. A Nonlinear Proportional Integral Derivative-Incorporated Stochastic Gradient Descent-based Latent Factor Model. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2020, Toronto, ON, Canada, 11–14 October 2020; pp. 2371–2376. [CrossRef]

30. Kalman, D. A Singularly Valuable Decomposition: The SVD of a Matrix. *Coll. Math. J.* **1996**, *27*, 2–23. [CrossRef]

31. Huang, K.; Sidiropoulos, N.D.; Swami, A. Non-Negative Matrix Factorization Revisited: Uniqueness and Algorithm for Symmetric Decomposition. *IEEE Trans. Signal Process.* **2014**, *62*, 211–224. [CrossRef]

32. Najariyan, M.; Zhao, Y. Granular fuzzy PID controller. *Expert Syst. Appl.* **2021**, *167*, 114182. [CrossRef]

33. Yu, H.; Guan, Z.; Chen, T.; Yamamoto, T. Design of data-driven PID controllers with adaptive updating rules. *Automatica* **2020**, *121*, 109185. [CrossRef]

34. Shekhar, S.; Xiong, H. Root-Mean-Square Error. In *Encyclopedia of GIS*; Shekhar, S., Xiong, H., Eds.; Springer US: Boston, MA, USA, 2008; pp. 979–979. [CrossRef]

35. Sammut, C.; Webb, G.I. (Eds.) Mean Absolute Error. In *Encyclopedia of Machine Learning and Data Mining*; Sammut, C.; Webb, G.I., Eds.; Springer US: Boston, MA, USA, 2017; pp. 806–806. [CrossRef]

36. Harper, F.M.; Konstan, J.A. The MovieLens Datasets: History and Context. *Acm Trans. Interact. Intell. Syst.* **2015**, *5*. [CrossRef]

37. Hug, N. Surprise: A Python library for recommender systems. *J. Open Source Softw.* **2020**, *5*, 2174. [CrossRef]