RLT: Residual-Loop Training in Collaborative Filtering for Combining Factorization and Global-Local Neighborhood

Lei Li[1,2], Weike Pan[1(✉)], Li Chen[2], and Zhong Ming[1(✉)]

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
lilei1995eli@gmail.com, {panweike,mingz}@szu.edu.cn
[2] Department of Computer Science, Hong Kong Baptist University, Hong Kong, China
lichen@comp.hkbu.edu.hk

Abstract. Collaborative filtering (CF) is an important recommendation problem focusing on predicting users' future preferences by exploiting their historical tastes. One typical training paradigm for this problem is called residual training (RT), which is usually built on two basic components of factorization- and local neighborhood-based methods in a sequential manner. RT has been well recognized with the ability of achieving higher recommendation accuracy than either factorization- or neighborhood-based method. In this paper, we design a new residual training paradigm called residual-loop training (RLT), which aims to fully exploit the complementarity of factorization, global neighborhood and local neighborhood in one single algorithm. Experimental results on three public datasets show the promising results of our RLT compared with several state-of-the-art methods.

Keywords: Residual training · Residual-loop training
Collaborative filtering

1 Introduction

Collaborative filtering (CF) is an important recommendation problem, where the main task is to exploit the historical (user, item, rating) triples and to predict the preferences w.r.t. (user, item) pairs not yet observed in the system. For this task, various techniques have been proposed, including factorization-based methods [7,9] and neighborhood-based methods [1]. There are also some training paradigms that are built on more than one basic model such as hybrid recommendation [5] and residual training [4]. Such training paradigms are usually reported with higher recommendation accuracy, especially in the context of international contests [3].

H. Jin et al. (Eds.): ICWS 2018, LNCS 10966, pp. 326–336, 2018.
https://doi.org/10.1007/978-3-319-94289-6_21

Residual training (RT) [4] has been well recognized as an effective preference learning framework in collaborative filtering. One of the most well-known paradigms in RT is "first factorization and then neighborhood" such as probabilistic matrix factorization (PMF) [9] followed by item-oriented collaborative filtering (ICF) [1]. The bridge between PMF and ICF is the residual, i.e., the difference between the true ratings of the training data and the predictions made by the factorization-based method, which is further exploited by the local neighborhood-based method, i.e., ICF. Residual training is usually more accurate than hybrid recommendation for preference prediction [4], which showcases the merit of the residual-based strategy of combining factorization- and local neighborhood-based methods as compared with the prediction-based ad-hoc strategy in hybrid methods.

We find that the traditional pipelined residual training paradigm may not be able to fully exploit the merits of factorization- and neighborhood-based methods. Firstly, there are two different types of neighborhood, i.e., global neighborhood in FISM [6] and SVD++ [7], and local neighborhood in ICF [1], but most residual training approaches ignore the global neighborhood. Secondly, combining the factorization-based method and neighborhood-based method in a pipelined residual chain may not be the best because the one-time interaction between the two methods may not be sufficient, but little research has been conducted on this issue.

In this paper, we propose a new residual training paradigm called residual-loop training (RLT). In RLT, we aim to combine factorization- and global-local neighborhood-based methods in a better way. Specifically, in our RLT, we adopt a different residual training strategy, i.e., "first factorization and global neighborhood, then local neighborhood, and finally factorization and global neighborhood again". More specifically, we put SVD++ [7] and ICF [1] in a loop instead of in a chain for achieving richer interactions between them, which is illustrated in Fig. 1.

We summarize our main contributions below: (i) we recognize the difference between global neighborhood and local neighborhood in the context of residual training; (ii) we propose to combine factorization-, global neighborhood-, and local neighborhood-based methods by residual training; and (iii) we propose a new residual training paradigm called residual-loop training (RLT). Extensive empirical studies on three public datasets show that our RLT can predict users' preferences more accurately.

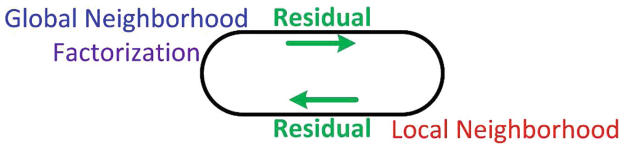

Fig. 1. Illustration of Residual-Loop Training (RLT).

2 Residual-Loop Training

2.1 Problem Definition

In our collaborative filtering problem, we have a set of (user, item, rating) triples as training data denoted by $\mathcal{R} = \{(u, i, r_{ui})\}$, where r_{ui} is the numerical rating assigned by user u to item i. Our goal is then to estimate the preference of user u to item j, i.e., $\hat{r}_{uj}$, for each record in the test data $\mathcal{R}^{te} = \{(u, i, r_{uj})\}$. Notice that the error defined on the difference between the predicted preference $\hat{r}_{uj}$ and the true preference r_{uj}, i.e., $\hat{r}_{uj} - r_{uj}$, will be used in the evaluation metric.

We put some commonly used notations in Table 1.

Table 1. Some notations.

u	User ID
i, i', j	Item ID
r_{ui}	Rating of user u to item i
$\mathcal{R} = \{(u, i, r_{ui})\}$	Rating records of training data
$\mathcal{U}_i$	Users who rate item i
$\mathcal{I}_u$	Items rated by user u
$\mathcal{N}_i$	Nearest neighbors of item i
$\mu \in \mathbb{R}$	Global average rating value
$b_u \in \mathbb{R}$	User bias
$b_i \in \mathbb{R}$	Item bias
$d \in \mathbb{R}$	Number of latent dimensions
$U_{u\cdot} \in \mathbb{R}^{1\times d}$	User-specific latent feature vector
$V_{i\cdot}, W_{i\cdot} \in \mathbb{R}^{1\times d}$	Item-specific latent feature vector
$\mathcal{R}^{te} = \{(u, j, r_{uj})\}$	Rating records of test data
$\hat{r}_{ui}$	Predicted rating of user u to item i
λ	Tradeoff parameter
T	Iteration number in the algorithm

2.2 Factorization-Based Method

Probabilistic matrix factorization (PMF) [9] is a seminal factorization-based method for rating prediction in collaborative filtering. Specifically, the prediction rule of the rating assigned by user u to item i is as follows,

$$\hat{r}^{\mathrm{F}}_{ui} = \mu + b_u + b_i + U_{u\cdot} V^T_{i\cdot}, \tag{1}$$

where μ, b_u and b_i are the global average, the user bias, and the item bias, respectively, and $U_{u\cdot} \in \mathbb{R}^{1\times d}$ and $V_{i\cdot} \in \mathbb{R}^{1\times d}$ are the user-specific latent feature vector and the item-specific latent feature vector, respectively.

Input: Users' rating records $\mathcal{R} = \{(u, i, r_{ui})\}$.

Output: Predicted preference of each record in the test data, i.e., $\hat{r}_{uj}$, $(u, j) \in \mathcal{R}^{te}$.

Task 1. Conduct factorization- and global neighborhood-based preference learning (i.e., SVD++), and estimate the preference of each record in the training data $\hat{r}_{ui}^{\text{F-N}g}$ and the preference of each record in the test data $\hat{r}_{uj}^{\text{F-N}g}$.

Task 2. Conduct local neighborhood-based preference learning (i.e., ICF) on the residual $r_{ui} - \hat{r}_{ui}^{\text{F-N}g}$, and estimate the preference of each record in the training data $\hat{r}_{ui}^{\text{N}\ell}$ and the preference of each record in test data $\hat{r}_{uj}^{\text{N}\ell}$.

Task 3. Conduct factorization- and global neighborhood-based preference learning again (i.e., SVD++) on the residual $r_{ui} - \hat{r}_{ui}^{\text{F-N}g} - \hat{r}_{ui}^{\text{N}\ell}$, and estimate the preference of each record in the test data $\hat{r}_{uj}^{\text{F-N}g\prime}$. Finally, the prediction of each record in the test data is obtained $\hat{r}_{uj} = \hat{r}_{uj}^{\text{F-N}g} + \hat{r}_{uj}^{\text{N}\ell} + \hat{r}_{uj}^{\text{F-N}g\prime}$.

Fig. 2. The algorithm of residual-loop training (RLT).

2.3 Local Neighborhood-Based Method

Item-oriented collaborative filtering (ICF) [1] is a classical neighborhood-based method for preference estimation in recommendation. The estimated preference of user u to item i can be written as follows,

$$\hat{r}_{ui}^{\text{N}\ell} = \sum_{i' \in \mathcal{I}_u \cap \mathcal{N}_i} \bar{s}_{i'i} r_{ui'}, \tag{2}$$

where $\bar{s}_{i'i} = s_{i'i} / \sum_{i' \in \mathcal{I}_u \cap \mathcal{N}_i} s_{i'i}$ is the normalized similarity with $s_{i'i} = |\mathcal{U}_{i'} \cap \mathcal{U}_i| / |\mathcal{U}_{i'} \cup \mathcal{U}_i|$ as the Jaccard index between item i' and item i. Notice that $\mathcal{N}_i$ is a set of locally nearest neighboring items of item i, i.e., their similarities are predefined via Jaccard index without global propagation among the users, and for this reason, we call it a local neighborhood-based method.

2.4 Global Neighborhood-Based Method

The similarity in Eq. (2) may also be learned from the data instead of being calculated from the training data. For example, in asymmetric factor model (AFM) [4], the prediction rule of user u to item i is as follows,

$$\hat{r}_{ui}^{\text{N}g} = \sum_{i' \in \mathcal{I}_u \setminus \{i\}} \bar{p}_{i'i}, \tag{3}$$

where $\bar{p}_{i'i} = W_{i'\cdot}V_{i\cdot}/\sqrt{|\mathcal{I}_u \backslash \{i\}|}$ is the normalized learned proximity between item i' and item i. Notice that $W_{i'\cdot}$ and $V_{i\cdot}$ are the item-specific latent feature vectors of item i' and item i, respectively. Considering the well-known merit of transitivity of latent factor models, the learned proximity in Eq. (3) is a global one because two items without common users may still be well connected via the learned latent factors. Notice that the prediction rule in Eq. (3) does not restrict to a local neighborhood set $\mathcal{N}_i$ as that in Eq. (2). We thus call AMF with the prediction rule in Eq. (3) a global neighborhood-based method.

2.5 Factorization- and Global Neighborhood-Based Method

Matrix factorization with implicit feedback (SVD++) [7] is a state-of-the-art method integrating the prediction rules of a factorization-based method and a global neighborhood-based method,

$$\begin{aligned} \hat{r}_{ui}^{\text{F-N}g} &= \mu + b_u + b_i + U_{u\cdot}V_{i\cdot}^T + \sum_{i' \in \mathcal{I}_u \backslash \{i\}} \bar{p}_{i'i}, \\ &= \hat{r}_{ui}^{\text{F}} + \hat{r}_{ui}^{\text{N}g}, \end{aligned} \tag{4}$$

from which we can see that SVD++ is a generalized factorization model that inherits the merits of both factorization- and global neighborhood-based methods.

2.6 Residual Training

Residual training (RT) [3,4] is an alternative approach to combining a factorization-based method and a neighborhood-based method. In RT, there are usually two steps with two different methods. Firstly, a factorization-based model is built using the training data, and a predicted rating $\hat{r}_{ui}^{\text{F}}$ for each $(u, i, r_{ui}) \in \mathcal{R}$ can then be obtained. Secondly, a neighborhood-based method is developed using $\sum_{i' \in \mathcal{I}_u \cap \mathcal{N}_i} \bar{s}_{i'i} r_{ui'}^{res}$, where $r_{ui'}^{res} = r_{ui'} - \hat{r}_{ui'}^{\text{F}}$ is the residual of the preceding factorization-based method. Notice that the Jaccard index $s_{i'i}$ is calculated using the residual data in our experiments, though it does not matter whether we use the residual data or the original training data because computing the Jaccard index $s_{i'i}$ only involves the IDs of the corresponding users.

We may represent the learning procedure as follows,

$$\hat{r}_{ui}^{\text{F}} \rightarrow \hat{r}_{ui}^{\text{N}\ell}, \tag{5}$$

where the bridge of the two preference learning methods is the "residual" as can be seen from the name of the method. The final prediction rule is then the summation of $\hat{r}_{ui}^{\text{F}}$ and $\hat{r}_{ui}^{\text{N}\ell}$, i.e., $\hat{r}_{ui}^{\text{F}} + \hat{r}_{ui}^{\text{N}\ell}$.

We can see that the main difference between SVD++ and RT are two folds: (i) SVD++ is an integrative method with one single prediction rule, while RT is a two-step approach with two separate prediction rules; and (ii) SVD++ exploits factorization and global neighborhood, while RT makes use of factorization and

local neighborhood. The merit of SVD++ is its modeling power using a complex prediction rule. As for RT, it is a flexible paradigm with two separate steps, which are of low dependency.

The above discussion motivates us to combine the merits of SVD++ and RT, and develop an improved algorithm accordingly, i.e., combining factorization, global neighborhood and local neighborhood, in one single algorithm.

2.7 Residual-Loop Training

In this paper, we aim to go one step beyond SVD++ that combines factorization- and global-neighborhood-based methods, and also residual training that combines factorization- and local neighborhood-based methods. Specifically, we would like to combine factorization, global neighborhood and local neighborhood in one single algorithm. For example, the final predicted preference of user u to item i should include $\hat{r}_{ui}^{\mathrm{F}}$ in Eq. (1), $\hat{r}_{ui}^{\mathrm{N}_\ell}$ in Eq. (2), and $\hat{r}_{ui}^{\mathrm{N}_g}$ in Eq. (3).

In order to fully exploit the complementarity of factorization, global neighborhood and local neighborhood, we propose a new residual training paradigm called residual-loop training (RLT), which is depicted as follows,

$$\hat{r}_{ui}^{\mathrm{F\text{-}N}_g} \rightarrow \hat{r}_{ui}^{\mathrm{N}_\ell} \rightarrow \hat{r}_{ui}^{\mathrm{F\text{-}N}_g} \tag{6}$$

where $\hat{r}_{ui}^{\mathrm{F\text{-}N}_g}$ is from Eq. (4) and $\hat{r}_{ui}^{\mathrm{N}_\ell}$ is from Eq. (2). We can see that our RLT in Eq. (6) is very different from RT in Eq. (5), which will be discussed in detail below.

For the first component in our RLT, i.e., $\hat{r}_{ui}^{\mathrm{F\text{-}N}_g}$ in Eq. (6), we aim to exploit both factorization and global neighborhood. The reason we adopt SVD++ instead of "first PMF and then AFM" in two separate steps, i.e., $\hat{r}_{ui}^{\mathrm{F}} \rightarrow \hat{r}_{ui}^{\mathrm{N}_g}$, is the close relationship between the factorization-based method in Eq. (1) and the global neighborhood-based method in Eq. (3). The interaction between the factorization-based method and the global neighborhood-based method is richer in such an integrative method than that in two separate steps of RT, i.e., $\hat{r}_{ui}^{\mathrm{F\text{-}N}_g}$ performs much better than $\hat{r}_{ui}^{\mathrm{F}} \rightarrow \hat{r}_{ui}^{\mathrm{N}_g}$, which is also observed in our preliminary empirical studies.

For the second component in our RLT, i.e., $\hat{r}_{ui}^{\mathrm{N}_\ell}$ in Eq. (6), we aim to boost the performance via local neighborhood. Notice that although $\hat{r}_{ui}^{\mathrm{F\text{-}N}_g}$ aims to integrate factorization and global neighborhood in one single model, but does not make a difference between global neighborhood and local neighborhood. Instead, we explicitly combine factorization, global neighborhood and local neighborhood for rating prediction in a residual-training manner. The performance is expected to be improved due to the complementarity, and the effectiveness of the residual training paradigm as verified in combining factorization and local neighborhood.

For the third component in our RLT, i.e., $\hat{r}_{ui}^{\mathrm{F\text{-}N}_g}$, we aim to further capture the remaining effects related to users' preferences that have not been modeled by the previous two methods yet. In the perspective of coarse-grained similarity calculation in neighborhood-based method and fine-grained parameter learning in model-based method, we use the first component again in this task, which results in a residual loop as shown in Fig. 1.

We depict the whole algorithm in Fig. 2. In Fig. 2, we can see that our RLT contains three tasks corresponding to the three components in the loop as shown in Eq. (6).

3 Experimental Results

3.1 Datasets and Evaluation Metric

In our empirical studies, we use three public datasets, including MovieLens 100K (ML100K), MovieLens 1M (ML1M) and MovieLens 10M (ML10M)[1]. We follow [8] and use 80% of each dataset as training data and the remaining 20% as test data, and repeat this for five times for five-fold cross validation.

We adopt the commonly used root mean square error (RMSE) in our performance evaluation, and report the average result from five-time evaluation.

3.2 Baselines and Parameter Settings

Our RLT is built on factorization-, global neighborhood- and local-neighborhood-based methods. We thus include the following closely related baseline methods to be compared with our RLT.

- Item-oriented collaborative filtering (ICF) [1] with Jaccard index as the similarity measurement.
- Probabilistic matrix factorization (PMF) [9].
- Hybrid collaborative filtering (HCF) [5] that averages the predictions of ICF and PMF, i.e., $\hat{r}_{ui} = (\hat{r}_{ui}^{ICF} + \hat{r}_{ui}^{PMF})/2$.
- Singular value decomposition with implicit feedback (SVD++) [7].
- Residual training (RT) [4] with PMF and ICF as two dependent components in a sequential manner.

For all factorization-based methods, we fix the number of latent dimensions as $d = 20$, the learning rate $\gamma = 0.01$, the iteration number as $T = 50$, and search the value of tradeoff parameters from $\{0.001, 0.01, 0.1\}$. For neighborhood-based methods, we take top-20 items from $\mathcal{I}_u \cap \mathcal{N}_i$ with highest Jaccard index as the neighbors. Notice that when $|\mathcal{I}_u \cap \mathcal{N}_i| < 20$, we use all items from $\mathcal{I}_u \cap \mathcal{N}_i$.

3.3 Results

We report the main results in Table 2. We can have the following observations:

- Our RLT predicts the users' preferences significantly more accurately than all other baseline methods, which clearly shows the advantage of our residual-loop training paradigm.

[1] http://grouplens.org/datasets/movielens/.

- For the performance of ICF, PMF and HCF, we can see that HCF improves the performance on ML100K and ML1M by combining the tentatively predicted preference of ICF and PMF, but not on ML10M, which shows the limitation of such a simple hybridization method.
- For the performance of ICF, PMF and SVD++, we can see that the performance ordering is ICF $<$ PMF $<$ SVD++, which shows the advantage of the factorization-based method (i.e., PMF) over the neighborhood-based method (i.e., ICF), and the further performance improvement by integrating factorization and neighborhood in one single method (i.e., SVD++).
- For the performance of ICF, PMF and RT, we can see that the performance ordering is ICF $<$ PMF $<$ RT, which shows the effectiveness of the residual training in exploiting the complementarity of the factorization-based method and the neighborhood-based method [3,4].
- For the performance of SVD++ and RT, we can see that their performance results are very close though the former exploits factorization and global neighborhood in an integrative way, and the latter exploits the factorization and local neighborhood in a pipelined manner, which also motivates us to further exploit the complementarity of factorization, global neighborhood, and local neighborhood.

Table 2. Recommendation performance of item-oriented collaborative filtering (ICF), probabilistic matrix factorization (PMF), hybrid recommendation combining ICF and PMF (HCF), SVD++, residual training (RT) and our residual-loop training (RLT). The significantly best results are marked in bold ($p < 0.01$). The values of the tradeoff parameter λ are also included for reproducibility.

	ML100K	ML1M	ML10M
ICF	0.9537 ± 0.0038	0.9093 ± 0.0021	0.8683 ± 0.0012
PMF	0.9441 ± 0.0038 ($\lambda = 0.01$)	0.8838 ± 0.0023 ($\lambda = 0.001$)	0.7911 ± 0.0005 ($\lambda = 0.01$)
HCF	0.9242 ± 0.0032 ($\lambda = 0.01$)	0.8739 ± 0.0023 ($\lambda = 0.001$)	0.8052 ± 0.0007 ($\lambda = 0.01$)
SVD++	0.9246 ± 0.0031 ($\lambda = 0.001$)	0.8515 ± 0.0018 ($\lambda = 0.001$)	0.7873 ± 0.0007 ($\lambda = 0.01$)
RT	0.9145 ± 0.0041 ($\lambda = 0.001$)	0.8567 ± 0.0021 ($\lambda = 0.001$)	0.7847 ± 0.0008 ($\lambda = 0.01$)
RLT	$\mathbf{0.8968} \pm 0.0040$ ($\lambda = 0.001$) ($\lambda = 0.001$)	$\mathbf{0.8385} \pm 0.0016$ ($\lambda = 0.001$) ($\lambda = 0.001$)	$\mathbf{0.7812} \pm 0.0007$ ($\lambda = 0.01$) ($\lambda = 0.01$)

We further study the performance of each task in our RLT, which is shown in Fig. 3. We can have the following observations:

- The performance improves in each subsequent task, e.g., "from SVD++ to ICF" and "from ICF to SVD++", in the residual loop of the algorithm shown

in Fig. 2, which shows the effectiveness of our residual-training mechanism that links factorization- and global-local neighborhood-based methods.
– The improvement "from SVD++ to ICF" is much larger than that "from ICF to SVD++", which implies that the second task is very useful while the third task is only marginally useful. This can be interpreted by the fact that the factorization and global-local neighborhood are somehow already well exploited in "SVD++ → ICF". Notice that although the further improvement in the third task of "from ICF to SVD++" is small, the improvement is still statistically significant.

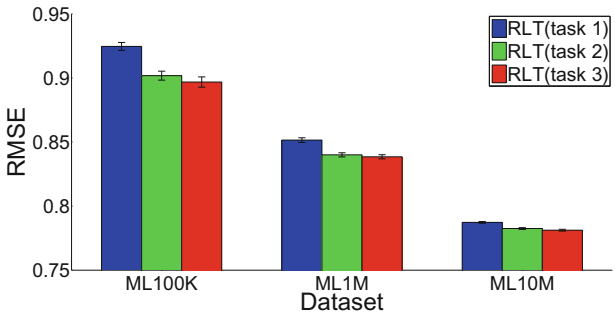

Fig. 3. Recommendation performance of three tasks in RLT, i.e., task 1 is SVD++, task 2 is ICF, and task 3 is SVD++ again.

4 Related Work

In this section, we study three categories of closely related work on improving one single collaborative filtering method, including hybridization, boosting and residual training.

4.1 Hybridization

In hybrid recommendation, the main idea is to combine the recommendation results of two or more different methods. The most popular hybridization strategy is probably to average the predicted ratings of some methods via certain weights [5]. For example, in our hybrid collaborative filtering (HCF), we combine the predictions of PMF and ICF via average weighting. Hybridization is simple and easy in real deployment, because the dependency among different methods is loose. However, this may also become the limitation of such a strategy, i.e., the complementarity of different methods may not be well exploited. Our empirical studies verify this point.

4.2 Boosting

In boosting-based recommendation, the main idea is to identify some difficult-to-learn (user, item, rating) triples in training data and then assign them higher weight and priority during the learning process [10]. Finally, a set of base models are learned with different weights on the triples. Those learned models are then combined via some weight based on the performance of each single model. Boosting has been well recognized as a very useful approach to boosting the performance of a single method. However, the dependency and complexity are much higher than those of the aforementioned hybridization and residual training framework studied in this paper.

4.3 Residual Training

In residual training [4], the main idea is to combine two different types of method via the residual of the prediction, which shows a close collaboration in making rating prediction and usually results in higher accuracy. Furthermore, the residual-based dependency is loose as compared with that of boosting.

We can see that the dependency becomes stronger from hybridization, residual training to boosting. And residual training achieves a good balance between the recommendation accuracy and the model dependency as compared with hybridization- and boosting-based recommendation methods.

5 Conclusions and Future Work

In this paper, we study the rating prediction problem in collaborative filtering by residual training. Specifically, we design a new residual training paradigm called residual-loop training (RLT), which aims to combine factorization, global neighborhood and local neighborhood in one single algorithm so as to fully exploit their complementarity. Experimental results on three public datasets show the significantly better performance of our RLT than several state-of-the-art factorization- and neighborhood-based methods.

For future work, we are interested in generalizing our residual-loop training to non-numerical ratings such as one-class feedback in E-commerce [2].

Acknowledgement. We thank the support of National Natural Science Foundation of China Nos. 61502307 and 61672358, Hong Kong RGC under the project RGC/HKBU12200415, and Natural Science Foundation of Guangdong Province No. 2016A030313038. Weike Pan and Zhong Ming are the corresponding authors for this work.

References

1. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. **22**(1), 143–177 (2004)
2. Huang, Z., Zeng, D., Chen, H.: A comparison of collaborative-filtering recommendation algorithms for e-commerce. IEEE Intell. Syst. **22**(5), 68–78 (2007)

3. Jahrer, M., Töscher, A.: Collaborative filtering ensemble. In: Proceedings of KDD Cup 2011 Competition, pp. 61–74 (2012)
4. Jahrer, M., Töscher, A., Legenstein, R.A.: Combining predictions for accurate recommender systems. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, pp. 693–702 (2010)
5. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction, 1st edn. Cambridge University Press, New York (2010)
6. Kabbur, S., Ning, X., Karypis, G.: FISM: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, pp. 659–667 (2013)
7. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 426–434 (2008)
8. Pan, W., Ming, Z.: Collaborative recommendation with multiclass preference context. IEEE Intell. Syst. **32**(2), 45–51 (2017)
9. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Annual Conference on Neural Information Processing Systems, NIPS 2008, pp. 1257–1264 (2008)
10. Wang, Y., Sun, H., Zhang, R.: AdaMF: adaptive boosting matrix factorization for recommender system. In: Li, F., Li, G., Hwang, S., Yao, B., Zhang, Z. (eds.) WAIM 2014. LNCS, vol. 8485, pp. 43–54. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08010-9_7