

Improving Diversity of User-Based Two-Step Recommendation Algorithm with Popularity Normalization

Xiangyu Zhao^{1,3}, Wei Chen^{2,4(✉)}, Feng Yang^{3,5}, and Zhongqiang Liu^{1,6}

¹ Beijing Research Center for Information Technology in Agriculture, Beijing, China

² Agricultural Information Institute, Chinese Academy of Agricultural Sciences,
Beijing, China
`chenwei@caas.cn`

³ National Engineering Research Center for Information Technology in Agriculture,
Beijing, China
`{zhaoxy,yangf}@nercita.org.cn`

⁴ Key Laboratory of Agri-information Service Technology,
Ministry of Agriculture, Beijing, China

⁵ Key Laboratory of Agri-informatics, Ministry of Agriculture, Beijing, China

⁶ Beijing Engineering Research Center of Agricultural Internet of Things,
Beijing, China
`liuqz@nercita.org.cn`

Abstract. Recommender systems become increasingly significant in solving the information overload problem. Beyond conventional rating prediction and ranking prediction recommendation technologies, two-step recommendation algorithms have been demonstrated that they have outstanding accuracy performance in top- N recommendation tasks. However, their recommendation lists are biased towards popular items. In this paper, we propose a popularity normalization method to improve the diversity of user-based two-step recommendation algorithms. Experiment results show that our proposed approach improves the diversity performance significantly while maintaining the advantage of two-step recommendation approaches on accuracy metrics.

Keywords: Recommender system · Collaborative filtering · Diversity · Two-step recommendation · Popularity normalization

1 Introduction

As the development of Internet and Mobile Internet, massive user-generated data give an opportunity to offer personalized information service with semantic computing [18, 25, 26]. Recommendation is an important kind of personalization technology which may help people in retrieving potentially useful information in a huge set of choices, especially in the current age of information overload. Collaborative filtering (CF) is a leading approach to build recommender systems which has gained much popularity recently [5, 6, 20, 21]. CF is based on the

analysis of past interactions between users and items, and hence can be readily applied in a variety of domains, without requiring external information about the traits of the recommended items.

Conventional CF approaches are based on users' rating values, for example from 1 to 5, and consider the recommendation problem as a rating prediction problem. These approaches estimate the ratings of items that have not been rated by the target user based on the rating history with a heuristic method [7, 9] or a learned model [4, 16, 31], and recommend top- N items with the highest predicted ratings. Therefore, many researchers focus on improving the prediction accuracy of unknown ratings. They consider that high quality of rating predictions directly indicates good recommendations [16, 31]. However, what people want from recommender systems is not whether the system can predict rating values accurately, but recommendations that match their interests [8]. Some researchers demonstrate that there is no trivial relationship between rating prediction accuracy and recommendation quality, as the rating prediction accuracy is not always consistent with ranking effectiveness [8, 19, 28]. Therefore, different from these rating prediction approaches, some researchers directly consider the recommendation problem as a ranking problem [17, 19, 24]. They propose models for ranking predictions by directly modeling user preferences with respect to a set of items rather than the rating scores on individual items.

We agree that the recommendation problem is a ranking problem. However, directly optimizing ranking targets may loss the semantic information of the recommendation scenario. Different from these ranking prediction CF, we proposed a two-step recommendation framework to generate recommendations by simulating the steps of users generating their rating data in previous work [28, 30]. Experiments show that our proposed approaches based on this framework gain better accuracy than conventional ones. However, beyond accuracy, other quality factors, such as diversity and novelty, are also important for recommendation technology [1, 3, 10, 12, 15, 23, 27, 32]. Some studies argue that one of the goals of recommender systems is to provide a user with highly idiosyncratic or personalized items, and more diverse recommendations result in more opportunities for users to get recommended such items [2]. In 2014, the ACM conference on Recommender Systems held an independent session "Diversity, Novelty and Serendipity"¹. Unfortunately, diversity and novelty of these two-step recommendation approaches are not very good. It appears that some recommendations of these approaches are biased towards well-known items, which may have been known by users. In this circumstance the recommendations are accurate, but not that valuable as the lack of novelty.

Therefore, the goal of this paper is to improve the diversity and novelty performance of two-step recommendation approaches while maintaining their advantages of accuracy. An improved user-based two-step recommendation algorithm with popularity normalization (UTSP) is proposed by adjusting the importance of items according to their popularity.

¹ <http://recsys.acm.org/recsys14/session-diversity-novelty-serendipity/>.

The remainder of the paper is organized as follows. Section 2 introduces two-step recommendation algorithm. UTSP is proposed in Sect. 3. Experiments are carried out on the MovieLens dataset in Sect. 4 to compare the proposed approach with some benchmark ones. We review related literature in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Two-Step Recommendation Algorithm

A typical Collaborative Filtering recommendation task is based on the rating data generated by users. These data contain two layers of user behaviors. The first one is that the current user selects an item to rate. The second one is rating it with a value. However, conventional recommendation approaches directly use rating values or rating ordinal relation to build rating prediction or ranking prediction algorithms and generate recommendations with no concern on the behaviors that users select items to rate. The effectiveness of these approaches is based on a condition that if a user rates an item, he/she may rate it with a value which is predicted by recommendation algorithms. Unfortunately, the prerequisite may not be satisfied. A user may not tend to rate an item, as it does not match his/her interest.

Therefore, we have proposed two-step recommendation algorithms to solve the recommendation problem in our previous work [28, 30]. In a two-step recommendation algorithm, the unknown user behaviors can be predicted as simulation of user generating ratings. That is predicting the probability $\hat{P}(u, i)$ that user u rates item i (in the first step), and then predicting the value $\hat{r}(u, i)$ which u may rate i with (in the second step). After that, the ranking score can be computed as:

$$ranking(u, i) = \hat{P}(u, i)\hat{r}(u, i) \quad (1)$$

This ranking score can be interpreted by the generation steps with a probability semantic. In addition, for a certain $\langle u, i \rangle$, the probability that the user may rate the item is $\hat{P}(u, i)$. Therefore, the probability that the user will not rate the item is $1 - \hat{P}(u, i)$. In recommender systems, typical values for the rated item are in 1–5 or 1–10 scale. In order to model rating values and rating behaviors in a unique model, the items that a user does not want to rate can be considered as being rated with value 0². In this way, the ranking score can be viewed as the mathematical expectation of users' rating on the items. This can be considered as another interpretation of the ranking score:

$$\begin{aligned} ranking(u, i) &= \hat{P}(u, i)\hat{r}(u, i) \\ &= \hat{P}(u, i)\hat{r}(u, i) + (1 - \hat{P}(u, i)) \cdot 0 \\ &= E[r(u, i)] \end{aligned} \quad (2)$$

² 0 is a typical value out of the range of rating scale, which can be used to distinguish the rating value and the rating behavior.

The goal of the first step is to predict the rating behaviors. Intuitively, historical rating behaviors are relevant to it, whereas rating values are not. Therefore, the probability is predicted using only rating behaviors in the first step of our proposed framework. In the second step, all users' historical rating data (both rating behaviors and rating values) are used to predict unknown ratings. This is a classical rating prediction problem. Existing techniques focus on rating prediction can be used in this step. After the two-step calculation, the ranking score can be computed with (1). The recommendation results can be generated based on the rankings.

3 UTSP Recommendation Algorithm

Recommender systems are explored to solve information overload problem for users. It means that the purpose of recommendation is inherently linked to a notion of discovery, as recommendation makes most sense when it exposes the user to a relevant experience that he/she would not have found by himself/herself. However, it is found that recommender systems actually can reduce the aggregate diversity, which has been described as “Harry Potter Problem” [11, 29]. Harry Potter is a runaway bestseller, which always appears in customers' recommendation list whatever books they are browsing. That is, recommended items are biased towards popular, well-known items. This can be explained by the fact that the idiosyncratic items often have limited historical data and, thus, are more difficult to be recommended to users; in contrast, popular items typically have more ratings and, therefore, can be recommended to more users. This phenomenon exists and is even worse in two-step recommendation algorithms, hence a UTSP algorithm is proposed to improve the diversity and novelty performance in this section.

3.1 The First Step

The target of the first step is to predict the probability that a user rates an item with users' historical rating behaviors. The rating behaviors are binary data, hence a user can be described as an n -dimensional vector in which 1 represents rated items and 0 represents unrated ones, which can be written as:

$$V_U(u) = (v_1, v_2, \dots, v_n) \quad (3)$$

$$v_i = \begin{cases} 1, & i \in I(u) \\ 0, & i \notin I(u) \end{cases} (i \in [1, n]) \quad (4)$$

where $I(u)$ represents the item set rated by user u .

Conventional user-based two-step recommendation algorithm (UTS) directly use this model to predict the probability that a user rates an item. If ignoring the effect of user similarity, the probability can be calculated as:

$$\hat{P}(u, i) = \frac{1}{|N(u)|} \sum_{a \in N(u)} V_U(a)[i] \quad (5)$$

³ <http://glinden.blogspot.com/2006/03/early-amazon-similarities.html>.

where $V_U(a)[i]$ is the i^{th} element of the binary user model for user a , $N(u)$ represents the neighbor set of user u , which contains the most similar users to user u .

This is the probability that an item rated by the neighbors for a given user. Intuitively, it is biased towards popular items as they have more ratings overall. Let's take the book domain for a motivate example. Harry Potter, as a bestseller, is bought by about 20 % of users, while Data Mining is a professional book for computer science researchers, only bought by no more than 0.3 % of total users. But in the neighbor set (with 50 neighbors) of a specific user a , there are 10 users who have bought Harry Potter, and 5 users who have bought Data Mining. If using (5) to generate recommendation directly, Harry Potter will be recommend to a . However, Data Mining might be a much better recommendation as the purchase rate in a 's neighbor set is far larger than the overall rate, the user may be a computer science researcher. It means that the increment of the purchase rate in a user's neighborhood is a more important measure than the value of purchase rate itself. It can be calculated as:

$$\hat{P}(u, i) = \frac{\sum_{a \in N(u)} V_U(a)[i] / |N(u)|}{|U(i)| / |U|} \quad (6)$$

where $U(i)$ represents the set of users who have rated item i , U represents the set of all the users. The value of the increased rate may be larger than 1, which is not suitable for the definition of probability. Moreover, since the main target of recommender systems is to recommend items for given users, and $|N(u)|$ and $|U|$ are constants for a given user, (6) can be simplified as:

$$\hat{P}(u, i) = \frac{\sum_{a \in N(u)} V_U(a)[i]}{|U(i)|} \quad (7)$$

As $U(i)$ is the popularity of item i , (7) can be considered as a normalization function with popularity. Furthermore, the attitude from a more similar user is always considered as more important information. Therefore, by including the similarity information, the probability that a user rates an item can be calculated as:

$$\hat{P}(u, i) = \frac{\sum_{a \in N(u)} sim(u, a) \cdot V_U(a)[i]}{|U(i)| \cdot \sum_{a \in N(u)} sim(u, a)} \quad (8)$$

where $sim(u, a)$ is the similarity between user u and user a .

In theory, this is an effective method to predict the probability that a user rates an item. However, according to our empirical study, the recommendations from (8) are biased towards long tail items. Let's get back to the book domain example. If only one neighbor has bought Data Mining because of his/her individual interest, the book still will be recommended to the user as its popularity is much less than Harry Potter. This means that the recommended items will be biased towards individual neighbor's long tail interest rather than the common

interest of the neighbor set. Therefore, we can decrease the degree of popularity normalization in order to reduce the bias towards long tail items. The revised function is written as:

$$\hat{P}(u, i) = \frac{\sum_{a \in N(u)} \text{sim}(u, a) \cdot V_U(a)[i]}{\beta \cdot \sqrt{|U(i)|} \cdot \sum_{a \in N(u)} \text{sim}(u, a)} \quad (9)$$

where β is a small constant to make sure the probability is between 0 and 1.

3.2 The Second Step

The second step is considered as a classical rating prediction problem. It can be done by making use of existing techniques. In UTSP, we use SVD++ [16] in the second step.

SVD++ is a matrix factorization approach, which is demonstrated to yield superior accuracy by considering implicit feedbacks⁴ as complement of explicit feedbacks (rating values), and using them together to build recommendation models by minimizing prediction errors. The prediction model of SVD++ is as follows:

$$\hat{r}(u, i) = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in Iu} y_j) \quad (10)$$

where μ is the average rating value of the known data. b_u and b_i indicate the observed deviations of user u and item i , respectively, from the average. p_u and q_i are the factorized user and item factor, respectively. y_j is an item factor which is computed according to the impact of implicit feedbacks.

SVD++ learns the values of involved parameters with a stochastic gradient descent technique by minimizing the regularized squared error function [16] associated with:

$$\begin{aligned} \min_{p^*, q^*, b^*, y^*} \sum_{\langle u, i \rangle} (r_{ui} - \mu - b_u - b_i - q_i^T (p_u + \\ |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j))^2 + \lambda_6 (b_u^2 + b_i^2) \\ + \lambda_7 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2) \end{aligned} \quad (11)$$

where r_{ui} is the actual rating value for item i rated by user u , λ_6 and λ_7 are two regularization parameters. The predicted ratings can be calculated by (10) using the learned parameters.

Based on the above models, UTSP can predict $\hat{P}(u, i)$ according to (9), predict $\hat{r}(u, i)$ according to (10), compute the rankings of the unrated items for users according to (1), and then generate recommendations.

⁴ Types of implicit feedback include rating behaviors, purchase history, browsing history, and search patterns.

4 Experiment

4.1 Experiment Setup

In this paper, we focus on both accuracy and diversity performance in top- N recommendation task, and use 4 metrics to evaluate our proposed approach. 1-call [22] and the Normalized Discounted Cumulative Gain (NDCG) [14] are used as accuracy metrics, whereas Coverage (COV) is used for evaluate the diversity of recommendations, and coverage in long tail (CIL) is mainly for evaluating novelty [30].

The proposed recommendation approach is evaluated on the MovieLens dataset, which consists of 100,000 ratings which are assigned by 943 users on 1682 movies. Collected ratings are in a 1-to-5 star scale. We use 5-fold cross validation for the evaluation. Starting from the initial data set, five distinct splits of training and test data are generated. For each data split, 80 % of the original set is included in the training data and 20 % of it is included in the test data. Users' rating history in the training set is used to generate recommendations according to different algorithms. The test set is then used to evaluate the recommendation results.

The proposed approach is compared with some benchmark ones for both rating prediction and ranking prediction approaches. For rating prediction approaches, UserCF [9] and SVD++ [16] are used for comparison. UserCF is a user-based CF with Jaccard as its similarity function. SVD++ is a state-of-the-art rating prediction approach. For ranking prediction approaches, pLPA [19] is used for comparison, which is a probabilistic latent preference analysis model which directly optimizes ranking target based on a pairwise ordinal model.

In addition, some approaches need user-specific parameters. The details of parameter assignments for different approaches are as follows: the size of nearest neighbors for UserCF is 50; SVD++ has 50 features and 25 iteration steps with $\lambda_6 = \lambda_7 = 0.05$, and $\gamma_1 = \gamma_2 = 0.002$; pLPA has 6 latent preferences and 30 iterations [19]. The first step of UTS and UTSP has the same neighbor size as UserCF, and the second step of UTS and UTSP has the same parameters as SVD++. Therefore, the effectiveness of the proposed approaches is irrelevant to the impact of these parameters.

With these parameters, all of the above mentioned approaches are evaluated by 1-call, NDCG, COV and CIL.

4.2 Experiment Results

In this subsection, UTSP will be compared with some benchmark recommendation approaches, including UserCF, SVD++, pLPA, and UTS, to demonstrate its effectiveness. For each approach, we report the NDCG values at the 1st, 3rd and 5th positions in the recommendation list, and 1-call, COV and CIL at the 5th position. Table 1 illustrates the results, where the top 2 best performed approaches for each metric have been highlighted.

Table 1. Performance of two-step recommendation approaches.

	NDCG			1-call	COV	CIL
	1	3	5			
UserCF	0.0101	0.0100	0.0131	0.0880	0.2949	0.1700
SVD++	0.0372	0.0448	0.0468	0.1994	0.0386	0.0065
pLPA	0.1506	0.1326	0.1211	0.3213	0.0428	0.0000
UTS	0.2082	0.1855	0.1750	0.4369	0.1034	0.0036
UTSP	0.1877	0.1690	0.1587	0.4358	0.2592	0.0910

As can be seen from the results, the two rating prediction approaches, UserCF and SVD++, get the worst accuracy. It means that there is no trivial relationship between the accuracy of rating prediction and quality of top- N recommendation. pLPA, a ranking prediction recommendation approach, can improve top- N recommendation accuracy from rating prediction approaches, which indicates that the recommendation problem is a ranking problem. For UTS and UTSP, these two-step recommendation approaches can further increase the recommendation accuracy. This shows that two-step recommendation is more suitable for top- N recommendation task.

Focusing on diversity metrics, UTS almost gets the worst performance. However, by using popularity normalization, UTSP improves its diversity performance significantly and achieves the 2nd best among all the approaches. If considering many recommended items from UserCF are irrelevant as its poor accuracy, UTSP recommends most effective items in terms of diversity. It means that UTSP outperforms all the benchmark recommendation approaches if considering both accuracy and diversity performance comprehensively.

5 Related Work

In this section, the review of literatures is divided into three parts. The first one is about conventional rating prediction recommendation algorithms. The second one includes some studies on ranking prediction recommendation approaches. The last one focuses on the two-step recommendation approaches.

5.1 Rating Prediction Approaches

Recommendation techniques have been studied for several years. Conventional recommendation approaches are based on rating prediction. In these approaches, the past interactions between users and items are analyzed by collaborative filtering. Algorithms of collaborative filtering can be divided into two classes: memory-based and model-based [2].

Memory-based algorithms are heuristic methods that make rating predictions based on the entire collection of items previously rated by users [7, 9]. They are

based on a basic assumption that people who agreed in the past tend to agree again in the future. The level of agreement can be measured by similarity. Based on the similarity calculation, recommender systems predict ratings for unknown items using adjusted weighted sum of known ratings and recommend items with high predicted values [9].

Model-based CF is another kind of typical CF methods. Model-based algorithms use the collection of ratings to learn a model, typically using some statistical machine-learning methods, which are then used to make rating prediction. These approaches always design appropriate loss functions and optimization procedures to learn their models by minimizing the error between predicted ratings and actual ones. Examples of such techniques include Bayesian clustering [4], matrix factorization [16, 31], and probabilistic Latent Semantic Analysis [13].

These conventional approaches are based on users' rating values, their optimization goals are minimizing prediction errors. Though they cannot generate top- N recommendation effectively, these techniques can be applied in the second step of two-step recommendation approaches.

5.2 Ranking Prediction Approaches

Differently from those rating prediction approaches, some researches directly consider the recommendation problem as a ranking prediction problem. They propose models for ranking prediction by directly modeling user preferences with respect to a set of items rather than the rating scores on individual items.

Weimer et al. [24] present a method (CofiRank) which uses Maximum Margin Matrix Factorization and considers maximum NDCG as the optimizing target. The approach is adaptable to different scores. Since the optimizing target of Cofirank is a listwise one, the approach scales well on collaborative filtering tasks.

Liu et al. [19] propose a probabilistic latent preference analysis (pLPA) model to make ranking predictions. From a user's observed ratings, they extract his/her preferences in the form of pairwise comparisons of items which are modeled by a mixture distribution based on Bradley-Terry model. An EM algorithm for fitting the corresponding latent class model as well as a method for predicting the optimal ranking is described.

Koren et al. [17] propose a collaborative filtering recommendation framework, which is based on the technique that considers user feedback on products as ordinal, rather than the more common numerical point of view. Their approach is based on a pointwise ordinal model, which allows it to linearly scale with data size. In addition, the approach can predict a full probability distribution of the expected item ratings, rather than only a single score for an item, and estimate the confidence level in each individual prediction.

It is demonstrated that these ranking prediction approaches can get better ranking results than rating prediction ones. However, experiments show that good performance on ranking prediction does not necessarily indicate good quality of top- N recommendation, which is the main purpose of recommender systems.

5.3 Two-Step Recommendation Approaches

Typical recommendation task is based on the rating data which contain two layers of user behaviors. The first one is that the current user selects an item to rate. The second one is rating it with a value. In this circumstance, simply using either rating prediction or ranking prediction idea to generate recommendations is ineffective since their basement condition that if a user rates an item, he/she may rate it with a value which is predicted by recommendation algorithms may not be satisfied. Therefore, two-step recommendation approaches try to solve recommendation problem in a different way.

Hofmann [13] decomposes the recommendation problem into the prediction of selected items and the prediction of the rating conditioned on the selected items. This mimics a scenario in which the user is free to select an item of his/her choice and also provides a rating for it. [28] finds that whether a user rates an item can be considered as a measure of interest no matter whether the value is high or low, and the rating values themselves represent the attitude to the quality of the target item, especially in the information overloaded age. Therefore, used-based and item-based two-step recommendation approaches are proposed by recommending items matching users' interests first, and then finding high quality items which users will like from the interested item set. [30] further proposes a two-step recommendation framework by simulating user generating ratings. That is predicting the probability that a user rates an item in the first step, and then predicting the value which the user may rate the item with in the second step. After that, the ranking score, which is used for generating recommendations, can be computed as the product of the probability and the value. Based on the framework, a hybrid approach of topic model and matrix factorization is proposed.

All the above two-step recommendation approaches gains good performance on accuracy in top- N recommendation task. The main difference between them is that Hofmann's approach is a intra two-step recommendation approach, which learns a unified model containing both steps, whereas others are inter two-step recommendation approaches, which combine two models, each of which processes in one step, respectively.

6 Conclusions

User-based two-step recommendation approach directly uses the probability that an item is rated by the neighbors to predict given user's possible rating behavior. It may cause recommendations to bias toward popular items.

By analyzing this problem, we propose a popularity normalization approach to improve UTS, which leads to significant diversity improvement while maintaining the good performance on accuracy. Experiment results show that our proposed approach outperforms the benchmark, including UserCF, SVD++, pLPA, and UTS while considering both accuracy and diversity performance comprehensively.

Acknowledgements. This work is supported by the National Key Technology R&D Program of China (project no. 2014BAD10B08).

References

1. Adamopoulos, P., Tuzhilin, A.: On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 153–160. ACM (2014)
2. Adomavicius, G., Kwon, Y.O.: Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* **24**(5), 896–911 (2012)
3. Bradley, K., Smyth, B.: Improving recommendation diversity. In: Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland, pp. 85–94. Citeseer (2001)
4. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)
5. Cai, Y., Lau, R.Y.K., Liao, S.S.Y., Li, C., Leung, H.-F., Ma, L.C.K.: Object typicality for effective web of things recommendations. *Decis. Support Syst.* **63**, 52–63 (2014)
6. Cai, Y., Leung, H., Li, Q., Min, H., Tang, J., Li, J.: Typicality-based collaborative filtering recommendation. *IEEE Trans. Knowl. Data Eng.* **26**(3), 766–779 (2014)
7. Chen, W., Niu, Z., Zhao, X., Li, Y.: A hybrid recommendation algorithm adapted in e-learning environments. *World Wide Web* **17**(2), 271–284 (2014)
8. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 39–46. ACM (2010)
9. Delgado, J., Ishii, N.: Memory-based weighted majority prediction. In: ACM SIGIR 1999 Workshop on Recommender Systems. Citeseer (1999)
10. Ekstrand, M.D., Harper, F.M., Willemsen, M.C., Konstan, J.A.: User perception of differences in recommender algorithms. In: Proceedings of the 8th ACM Conference on Recommender systems, pp. 161–168. ACM (2014)
11. Fleder, D., Hosanagar, K.: Blockbuster culture’s next rise or fall: The impact of recommender systems on sales diversity. *Manage. Sci.* **55**(5), 697–712 (2009)
12. Garcin, F., Faltings, B., Donatsch, O., Alazzawi, A., Bruttin, C., Huber, A.: Offline and online evaluation of news recommender systems at swissinfo.ch. In: Proceedings of the 8th ACM Conference on Recommender systems, pp. 169–176. ACM (2014)
13. Hofmann, T.: Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst. (TOIS)* **22**(1), 89–115 (2004)
14. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst. (TOIS)* **20**(4), 422–446 (2002)
15. Kapoor, K., Kumar, V., Terveen, L., Konstan, J.A., Schrater, P.: I like to explore sometimes: Adapting to dynamic user novelty preferences. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 19–26. ACM (2015)
16. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative-filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
17. Koren, Y., Sill, J.: OrdRec: An ordinal model for predicting personalized item rating distributions. In: Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 117–124. ACM (2011)

18. Li, X., Xie, H., Song, Y., Li, Q., Shanfeng Zhu, F., Wang, L.: Does summarization help stock prediction? News impact analysis via summarization. *IEEE Intell. Syst.* **30**, 26–34 (2015)
19. Liu, N.N., Zhao, M., Yang, Q.: Probabilistic latent preference analysis for collaborative filtering. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 759–766. ACM (2009)
20. Perugini, S., Gonçalves, M.A., Fox, E.A.: Recommender systems research: A connection-centric survey. *J. Intell. Inf. Syst.* **23**(2), 107–143 (2004)
21. Ricci, F., Shapira, B.: *Recommender Systems Handbook*. Springer, Heidelberg (2011)
22. Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Oliver, N., Hanjalic, A.: CliMF: Learning to maximize reciprocal rank with collaborativeless-is-more filtering. In: *Proceedings of the Sixth ACM Conference on Recommender Systems*, pp. 139–146. ACM (2012)
23. Vargas, S., Castells, P.: Improving sales diversity by recommending users to items. In: *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 145–152. ACM (2014)
24. Weimer, M., Karatzoglou, A., Le, Q.V., Smola, A.J.: Cofi rank-maximum margin matrix factorization for collaborative ranking. In: *Advances in Neural Information Processing Systems*, pp. 1593–1600 (2007)
25. Xie, H.-R., Li, Q., Cai, Y.: Community-aware resource profiling for personalized search in folksonomy. *J. Comput. Sci. Technol.* **27**(3), 599–610 (2012)
26. Xie, H., Yu, L., Li, Q.: A hybrid semantic item model for recipe search by example. In: *IEEE International Symposium on Multimedia (ISM)*, pp. 254–259. IEEE (2010)
27. Zhang, M., Hurley, N.: Avoiding monotony: Improving the diversity of recommendation lists. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*, pp. 123–130. ACM (2008)
28. Zhao, X., Niu, Z., Chen, W.: Interest before liking: Two-step recommendation approaches. *Knowl. Based Syst.* **48**, 46–56 (2013)
29. Zhao, X., Niu, Z., Chen, W.: Opinion-based collaborative filtering to solve popularity bias in recommender systems. In: Decker, H., Lhotská, L., Link, S., Basl, J., Tjoa, A.M. (eds.) *DEXA 2013, Part II. LNCS*, vol. 8056, pp. 426–433. Springer, Heidelberg (2013)
30. Zhao, X., Niu, Z., Chen, W., Shi, C., Niu, K., Liu, D.: A hybrid approach of topic model and matrix factorization based on two-step recommendation framework. *J. Intell. Inf. Syst.* **44**(3), 335–353 (2014)
31. Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-scale parallel collaborative filtering for the netflix prize. In: Fleischer, R., Xu, J. (eds.) *AAIM 2008. LNCS*, vol. 5034, pp. 337–348. Springer, Heidelberg (2008)
32. Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: *Proceedings of the 14th International Conference on World Wide Web*, pp. 22–32. ACM (2005)

Database Systems for Advanced Applications

DASFAA 2016 International Workshops: BDMS, BDQM,
Mol, and SeCoP, Dallas, TX, USA, April 16-19, 2016,
Proceedings

Gao, H.; Kim, J.; Sakurai, Y. (Eds.)

2016, XIII, 394 p. 121 illus., Softcover

ISBN: 978-3-319-32054-0