Improving the Accuracy of Latent-space-based Recommender Systems by Introducing a Cut-off Criterion

Ludovico Boratto,
Salvatore Carta,
Roberto Saia
Department of Mathematics and
Computer Science
University of Cagliari
Via Ospedale 72, 09124,
Cagliari, Italy
{ludovico.boratto, salvatore,
roberto.saia}@unica.it

EICS'16, June 21-24, 2016, Bruxelles, Belgium.

Abstract
Recommender systems filter the items a user did not evaluate, in order to acquire knowledge on the those that might be suggested to her. To accomplish this objective, they employ the preferences the user expressed in forms of explicit ratings or of implicitly values collected through the browsing of the items. However, users have different rating behaviors (e.g., users might use just the ends of the rating scale, to expressed whether they loved or hated an item), while the system assumes that the users employ the whole scale. Over the last few years, *Singular Value Decomposition* (SVD) became the most popular and accurate form of recommendation, because of its capability of working with sparse data, exploiting latent features. This paper presents an approach that pre-filters the items a user evaluated and removes those she did not like. In other words, by analyzing a user's rating behavior and the rating scale she used, we capture and employ in the recommendation process only the items she really liked. Experimental results show that our form of filtering leads to more accurate recommendations.

Author Keywords
Data Mining; Recommender Systems; User Profiling; Algorithms.

Introduction

A recommender system is designed to suggest items of possible interest to the users [24]. In order to generate the recommendations, different forms of data are employed by the different types of systems. Indeed, the two most effective classes of systems, i.e., *collaborative filtering* and *content-based* approaches, respectively use (i) the ratings given by the user to express a preference for the items and (ii) the content of the items (e.g., their textual description).

Independently from the employed approach, user ratings (or implicitly collected values, like the number of seconds spent while browsing an item's characteristics) are the elements that allow a system to acquire knowledge on what the users like, or not. However, it is widely-known that users have different rating behaviors and that some of them do not use the whole rating scale, but express only whether they love or hate an item [21].

At the moment, however, all the recommender systems base their filtering on a unique scale of values. Therefore, if a user is required to express a rating in a defined scale, the system assumes that the rating behavior of the user covers the whole scale. Instead, if the system implicitly collects the data, a fixed cut-off value is chosen, in order to determine if a user liked an item (e.g., Fastweb's recommender system collects a positive preference for a TV program if a user watches it for at least 5 minutes [5]).

It is widely-known that the recommendation form that generates the most accurate results is collaborative filtering and, more specifically, it is Koren's implementation of *Singular Value Decomposition* (SVD), known as $SVD++$ [15]. The algorithm is able to find latent spaces, based on the ratings expressed by the users, thus avoiding problems such as sparsity and improving the efficiency of the algorithm.

The problem that might arise is that if users have different behaviors (both when rating the items and when browsing the Web), the system might consider as liked by a user an item with a high rating, but that actually represents the lowest rating she gave (the same problem holds in the opposite scenario, in which a user only gives low ratings and her favorite item might be misclassified if considering the system's scale).

The intuition behind this paper is that, since SVD can detect latent spaces and work with sparse data, the algorithm might benefit from receiving less *but very accurate* information about what the users actually like.

In this work, we first show that users have different ratings behaviors, then we propose an approach that calculates the weighted average of the user ratings and leaves in the user profile only the ratings greater or equal than this value, thus removing the other ones. By modeling the positive behavior of the users and understand what they actually like, our approach should lead to more accurate recommendations. Note that this study is based on explicitly given ratings to facilitate its validation with a public dataset, but this technique can be applied, as is, to implicitly-collected data (e.g., by removing all the items that have been browsed for a number of seconds lower than the user's average).

More formally, the problem statement is the following:

Problem 1 *We are given a set of users $U = \{u_1, \ldots, u_N\}$, a set of items $I = \{i_1, \ldots, i_M\}$, and a set $R = [1, 5]$ of ratings used to express the user preferences. The set of all possible preferences expressed by the users is a ternary relation $P \subseteq U \times I \times R$. We also denote as $I_u = \{i \in I | \exists (u, i, r) \in P \wedge u \in U\}$ the set of items in the profile of a user u. Let SVD_{I_u} denote the fact that the SVD algorithm is run with the set of preferences $I_u, \forall u \in U$. Our*

objective is to define a Weighted Cut-off Criterion *(WCC) able to generate a set $\hat{I}_u$, which considers the rating scale employed by the user and removes from the set of items positively evaluated by her (I_u) those in the lower part of her scale. The goal of this paper is to show that* $accuracy(SVD_{\hat{I}_u}) > accuracy(SVD_{I_u})$.

The contributions of our work are reported in the following:

- analysis of the user ratings of a real-world dataset, aimed to show the non-coincidence of the range of values adopted by the users to rate the evaluated items, with that defined by the recommender system;
- formalization of a Weighted Cut-off Criterion (WCC) able to remove from the user ratings those below the weighted mean value of her preferences;
- evaluation of the proposed criterion, by comparing the performance of a state-of-the-art recommendation approach, before and after the WCC process applied to the user ratings.

In the rest of this paper, we first show that users actually have different rating behaviors (Section “Analysis of the Users’ Rating Behavior"), continuing by defining our approach (Section “Approach"). Then we present the results of the performed experiments (Section “Evaluation"), the literature related with our study (Section “Background and Related Work"), concluding with some remarks (Section “Conclusions and Future Work").

Analysis of the Users’ Rating Behavior

In order to validate our intuition and understand if users actually have different rating behaviors or if they use the whole rating scale, in this section we are going to present the number of users who use a specific rating scale.

The study has been performed on the Yahoo! Webscope R4[1] dataset. It contains a large amount of data related to users preferences expressed by the Yahoo! Movies community that are rated on the base of two different scales, from 1 to 13 and from 1 to 5 (we have chosen to use the latter). The training data is composed by 7,642 users ($|U|$), 11,915 movies/items ($|I|$), and 211,231 ratings ($|R|$), and all users involved have rated at least 10 items and all items are rated by at least one user. The test data is composed by 2,309 users, 2,380 items, and 10,136 ratings. There are no test users/items that do not also appear in the training data. Each user in the training and test data is represented by a unique ID.

Figure 1 shows that the users express their ratings in different ways with respect to the range of values allowed by the recommender system (in our case, we have $R = [1, 5]$).

Since the users in the dataset are $7,642$, only about half of them (52.51%, $4,013$ users) have given their opinion by using the whole rating scale, while the others have used a different range of values. Indeed, these users can be mostly classified in three groups: $1,319$ users (17.25%) that used the range $3 \div 5$ (i.e., by evaluating their worst experience with a minimum score of 3), $1,315$ users (17.20%) that used the range $2 \div 5$ (i.e., by evaluating their worst experience with a minimum score of 2), and 688 users (9.00%) that expressed their opinion in the range $4 \div 5$ (i.e., by keeping an high rating in all their evaluations).

These results clearly indicate that each user adopts personal criteria of evaluation. For this reason, an effective exploitation of her preferences should take into account this aspect.

[1] http://webscope.sandbox.yahoo.com

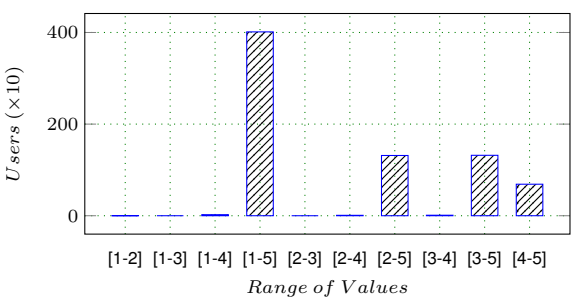

Figure 1: $Ranges\ of\ Values\ in\ User\ Evaluations$

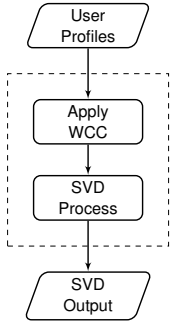

Figure 2: Approach Architecture

Approach

Given the fact that users have different rating behaviors, in this section we present an algorithm that detects the rating scale of a user and removes from the items she evaluated those under her average rating. The algorithm performs two main steps:

- **Weighted Cut-off Criterion**. Calculation of the average value of the ratings of each user (*Weighted Ratings Average*) and definition of a *Weighted Cut-off Criterion* (WCC), to keep only the items with a rating above the user's average.
- **Item Recommendation**. The state-of-the-art algorithm SVD is run with the items processed by the previous step.

The architecture of the proposed system is summarized in Figure 2. In the following, we will describe in detail how each step works.

Weighted Cut-off Criterion

In order to understand which item a user actually likes, it is first necessary to identify the average rating, among those she gave. This value, called *Weighted Ratings Average* (WRA), is calculated on the basis of the ratings of each user $u \in U$, in the context of her profile $i \in I_u$, as shown in Equation 1.

$$WRA(u) = \frac{\sum_{i \in I_u} r}{\mid I_u \mid} \tag{1}$$

Given the WRA of a user, we define the *Weighted Cut-off Criterion* (WCC) that allows us to filter the user ratings $r \in R$ in the profile I_u of a user $u \in U$. We perform this operation for each item $i \in I_u$, according to the criterion shown in Equation 2.

$$r = \begin{cases} 0, & \text{if } r < WRA(u) \\ r, & \text{otherwise} \end{cases} \tag{2}$$

The output of this step is a set $\hat{I}_u$, which contains the ratings processed through the WCC criterion.

This filtering process is summarized in Algorithm 1.

Algorithm 1 $Ratings\ filtering$

Input: I_u=User profile
Output: $\hat{I}_u$ = Filtered user profile
1: **procedure** FILTERUSERPROFILE(I_u)
2: WRA=GetWeightedRatingsAverage(I_u)
3: **for** each i in I_u **do**
4: r=GetItemRating(i)
5: **if** $r \geq WRA$ **then**
6: $\hat{I}_u \leftarrow (i, r)$
7: **end if**
8: **end for**
9: Return $\hat{I}_u$
10: **end procedure**

The algorithm takes as input (*step 1*) the profile I_u of a user $u \in U$, and provides as output (*step 9*) this user profile $\hat{I}_u$ filtered on the basis of the proposed WCC criterion. After the calculation of the Weighted Ratings Average (WRA) of the items in the user profile I_u, performed at the *step 2*, from the *step 3* to *step 8* we extract the rating r of each item $i \in I_u$ (*step 4*), by adding it to the set $\hat{I}_u$, when the value of R is greater or equal the WRA value (from *step 5* to *step 6*). The set $\hat{I}_u$, that represents the user profile I_u filtered on the basis of the proposed criterion, is returned as output at the end of the process (*step 9*).

Item Recommendation
This step runs the state-of-the-art SVD algorithm, by using as input the set $\hat{I}_u$, for each user $u \in U$. In that way, the algorithm only processes the items for which a user expressed an interest above the average.

Evaluation

In this section, we first describe the environment and the parameters setup, then we present the strategy and the involved metric, concluding with the experimental results and their discussion. The dataset employed to validate our proposal is the same one previously presented, i.e., Yahoo! Webscope (R4).

Environment
The environment is based on the Java language, with the support of the Mahout framework[2] to implement the state-of-the-art recommendation approach (i.e., SVD) and to perform the evaluation of the experimental results in terms of accuracy. The experimental framework was developed by using a machine with an Intel i7-4510U, quad core (2 GHz $\times$ 4) and a Linux 64-bit Operating System (*Debian Jessie*) with 4 GBytes of RAM.

Parameters Setup
The optimal values of two of the three parameters needed to run the Mahout implementation of SVD (i.e., the regularization parameter $lambda$ used to avoid overfitting and the number of *training steps*) have been chosen through a preliminary training (the selected values are, respectively, 0.05 and 10). The third parameter (i.e., the dimension of the feature space) was instead tested in a range from 2 to 20, during the set of experiments aimed to evaluate the accuracy of the SVD recommender approach, before and after the use of our WCC approach. This is useful to test the effectiveness of the proposed approach when the size of the latent space varies.

Metric
The accuracy of the performed recommendations was measured through the Root Mean Squared Error ($RMSE$). This metric considers the test set and the predicted ratings by comparing each rating r_{ui}, given by a user u for an item i and available in the test set, with the rating p_{ui} predicted by a recommender system. Its formalization is shown in the Equation 3, where n is the number of ratings available in the

[2] http://mahout.apache.org/

test set.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(r_{ui} - p_{ui})^2}{n}} \quad (3)$$

Strategy
We evaluate our proposal through a comparative analysis, by considering the recommendations generated by the SVD approach, before and after the proposed filtering process, based on the WCC criterion. The comparisons have been made by measuring the results accuracy through the well-known *Root Mean Squared Error* ($RMSE$) metric, described in the previous Section “Metric". In order to guarantee the repeatability of the performed experiments, according with the Mahout documentation we used in the Java code the instruction *RandomUtils.useTestSeed()*. The evaluation process has been performed by using the Mahout functionalities designed to perform this task (*RecommenderEvaluator* Java class).

We validate our proposal by running a set of experiments that measure the accuracy of the SVD recommendations, before and after the use of our WCC approach.

Experimental Results
As shown in Figure 3, our approach gets better accuracy values along almost all the considered SVD feature space range. It means that a preliminary filtering of the user ratings leads toward a better performance in approaches of recommendation such as SVD, which are strongly based on this kind of information. We can observe how the $RMSE$ values get worse when the latent space increases (SVD_{I_u} approach), while they remain stable in our case ($SVD_{\hat{I}_u}$ approach), showing that the accuracy of the

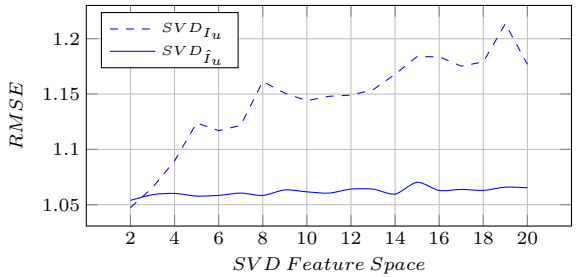

Figure 3: $Recommendations\ Accuracy$

system worsens when increasing the latent space, if has been used non-filtered user ratings.

Results Summary
The results obtained in this study first showed the existence of the problem related with the different ways that the users adopt to evaluate the items, while the second illustrates how the proposed WCC approach is able to improve the performance of the state-of-the-art recommendation approach, SVD. Indeed, the results of the experiments show us that the range of values that the users adopt during their evaluations, are in the half of the cases different from that allowed by the system (Figure 1), and that a preliminary filtering of them by our *Weighted Cut-off Criterion* overcomes this problem, improving the accuracy of the recommendations (Figure 3).

Background and Related Work

In this section we briefly review some main concepts closely related with the proposed work.

User Profiling. In the e-commerce environment the recommender systems play a determinant role, their first implementations were based on the so-called *Collaborative*

Filtering approach [13, 14], which is based on the assumption that users have similar preferences on a item, if they already have rated other similar items [27]. An alternative approach is that defined as *Content-based*, where the items to recommend are those whose content is similar to that of the items previously evaluated by the user [19, 22]. The early systems used relatively simple retrieval models, such as the Vector Space Model, with the basic TF-IDF weighting [4, 6, 18, 23], a spatial representation of the textual description of the items, where each of them is represented by a vector in a $n\text{-}dimensional$ space, and each dimension is related to a term from the overall vocabulary of a specific document collection.

There are several approaches to create user profiles: some of them focus on *short-term* user profiles that capture features of the user's current search context [7, 11, 26], while others accommodate *long-term* profiles that capture the user preferences over a long period of time [3, 8, 20]. As shown in [28], compared with the *short-term* user profiles, the use of a *long-term* user profile generally produces more reliable results, at least when the user preferences are fairly stable over a long time period. It should be noted that, regardless of the approach used to define the user profiles, almost all the state-of-the-art strategies take into account, as main source of information, the user ratings (i.e., the score given to the evaluated items by them), or by using directly them, or by exploiting their latent characteristics (e.g., latent-factor-based [15]).

Latent Factor Models. The type of data with which a recommender system operates is typically a sparse matrix where the rows represent the users, and the columns represent the items. The entries of this matrix are the interactions between users and items, in the form of ratings or purchases. The aim of a recommender system is to infer, for each user u, a ranked list of items, and in literature many of them are focused on the rating prediction problem. The most effective strategies in this field exploit the so-called *latent factor models*, but especially, the *matrix factorization* techniques [16]. Other CF ranking-oriented approaches that extend the matrix factorization techniques, have been recently proposed, and most of them use a ranking oriented objective function, in order to learn the latent factors of users and items [17]. Nowadays, the *Singular Value Decomposition* (SVD) [10] approach and its Koren's version $SVD++$ [15] are considered the best strategies in terms of accuracy and scalability.

User Ratings Reliability. The concept of bias, introduced in a recommender system process as noise in user ratings, is well known in literature since 1995, when it was cited in a work aimed at discussing the concept of reliability of users in terms of rating coherence [13]. Similar studies have been performed subsequently, such as that in [9], where hundreds of users evaluated a set of movies, randomly selected, which they have already evaluated in the past, with the result to show an incoherence in their evaluations in the 40% of cases. All these studies lead toward the same problem that in literature is identified as *magic barrier* [12], a term used to define the theoretical boundary for the level of optimization that can be achieved by a recommendation algorithm on transactional data [25]. The evaluation models assume as a ground truth that the transactions made in the past by the users, and stored in their profiles, are free of noise. This is a concept that has been studied in [2, 1], where a study aimed to capture the noise in a service that operates in a synthetic environment was performed.

To the best of our knowledge, there are not studies aimed to tackle the problem of the inconsistence in the user ratings, when this issue derives from the different ways adopted by

the users to assign a rating to the evaluated items. The approach proposed in this work aimed at addressing the aforementioned problem in a twofold manner: first, it wanted to define a method able to operate with any type of profile (e.g., *short-term* or *long-term* profiles); second, it wanted to face the limitation related with the magic barrier problem, by removing from the user profiles all the ratings that do not reflect the user preferences in terms of weighted ratings average, i.e., those items that could represent a kind of noise in the recommender process.

Conclusions and Future Work

The work presented in this paper wanted to highlight and face a problem that rises when the users assign a rating to the evaluated items, by adopting a range of values that could not cover the entire interval allowed by the system with which they interact. Through the first experiment we showed the real existence of this problem, which has been faced by introducing a novel cut-off criterion (WCC). This criterion is based on a weighted ratings average value (WRA), and its effectiveness to improve the accuracy of a recommender system at the state of the art, such a SVD, has been demonstrated in the second experiment.

In summary, our contribution in this field is twofold: first, we are able to improve the performance of a recommender system based on the user ratings (almost all of the state-of-the-art approaches); second, we are also able to reduce the computational load of the recommendation process, thanks to the removal of certain items from the user profiles (i.e., those with a rating less than WRA value). It should also be noted how the proposed approach leads toward a considerable improvements of the accuracy of the recommendations, without the needing to adopt sophisticated mathematical criteria to preprocess the user ratings, with a great advantage in terms of computational complexity.

Future work will study the relations between the range of values adopted by the users to express their opinion in different domains, to model in a better way the preferences in environments that sell different types of items. (e.g., a site such as Amazon[3], which sells different types of goods). Indeed, each type of item might be associated to a different rating behavior. This will allow us to generate more effective recommendations.

Acknowledgment

This work is partially funded by Regione Sardegna under project NOMAD (Next generation Open Mobile Apps Development), through PIA - Pacchetti Integrati di Agevolazione “Industria Artigianato e Servizi" (annualità 2013).

References

[1] Xavier Amatriain, Josep M Pujol, and Nuria Oliver. 2009a. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *User modeling, adaptation, and personalization*. Springer, 247–258.

[2] Xavier Amatriain, Josep M Pujol, Nava Tintarev, and Nuria Oliver. 2009b. Rate it again: increasing recommendation accuracy by user re-rating. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 173–180.

[3] Fabio A Asnicar and Carlo Tasso. 1997. ifWeb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In *Sixth International Conference on User Modeling*. 2–5.

[4] Marko Balabanović and Yoav Shoham. 1997. Fab: content-based, collaborative recommendation.

[3] http://www.amazon.com/

Commun. ACM 40, 3 (1997), 66–72.

[5] Riccardo Bambini, Paolo Cremonesi, and Roberto Turrin. 2011. A Recommender System for an IPTV Service Provider: a Real Large-Scale Production Environment. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer, 299–331.

[6] Daniel Billsus and Michael J Pazzani. 1999. *A hybrid user model for news story classification*. Springer.

[7] Jay Budzik and Kristian J. Hammond. 2000. User Interactions with Everyday Applications As Context for Just-in-time Information Access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI '00)*. ACM, New York, NY, USA, 44–51.

[8] Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. 2005. Using ODP metadata to personalize search. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 178–185.

[9] Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl. 2003. Is seeing believing?: how recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 585–592.

[10] Michael J. Pazzani Daniel Billsus. 1998. Learning Collaborative Information Filters. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, Jude W. Shavlik (Ed.). Morgan Kaufmann, 46–54.

[11] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing Search in Context: The Concept Revisited. *ACM Trans. Inf. Syst.* 20, 1 (Jan. 2002), 116–131.

[12] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (2004), 5–53.

[13] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. 1995. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 194–201.

[14] George Karypis. 2001. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 247–254.

[15] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[16] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[17] Yehuda Koren and Joe Sill. 2011. OrdRec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 117–124.

[18] Henry Lieberman and others. 1995. Letizia: An agent that assists web browsing. *IJCAI (1)* 1995 (1995), 924–929.

[19] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

[20] Zhongming Ma, Gautam Pant, and Olivia R. Liu Sheng. 2007. Interest-based Personalized Search. *ACM Trans. Inf. Syst.* 25, 1, Article 5 (Feb. 2007).

[21] Xia Ning, Christian Desrosiers, and George Karypis. 2015. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 37–76.

[22] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[23] Michael J Pazzani, Jack Muramatsu, Daniel Billsus, and others. 1996. Syskill & Webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*. 54–61.

[24] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*. Springer US, 1–34.

[25] Alan Said, Brijnesh J Jain, Sascha Narr, Till Plumbaum, Sahin Albayrak, and Christian Scheel. 2012. Estimating the magic barrier of recommender systems: a user study. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1061–1062.

[26] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 824–831.

[27] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.

[28] Dwi H Widyantoro, Thomas R Ioerger, and John Yen. 2001. Learning user interest dynamics with a three-descriptor representation. *Journal of the American Society for Information Science and Technology* 52, 3 (2001), 212–225.