CBRec: A Book Recommendation System for Children Using the Matrix Factorization and Content-Based Filtering Approaches[*]

Yiu-Kai Ng

Computer Science Department
Brigham Young University
Provo, Utah, 84602, U.S.A.

ng@compsci.byu.edu

Abstract. Promoting good reading habits among children is essential, given the enormous influence of reading on students' development as learners and members of the society. Unfortunately, very few (children) websites or online applications recommend books to children, even though they can play a significant role in encouraging children to read. Given that a few popular book websites suggest books to children based on the popularity of books or rankings on books, they are not customized/personalized for each individual user and likely recommend books that users do not want or like. We have integrated the Matrix Factorization approach and the content-based approach, in addition to predicting the grade levels of books, to recommend books for children. Recent research works have demonstrated that a hybrid approach, which combines different filtering approaches is more effective in making recommendations. Conducted empirical study has verified the effectiveness of our proposed children book recommendation system.

Keywords: Book recommendation, matrix factorization, content analysis, children

1 Introduction

Recommender systems are available and widely used these days. Services that use recommender systems include Amazon, Netflix, and Youtube. Various recommender systems adopt different algorithms to suggest items for the users. For examples, item-based and user-based Collaborative filtering (CF) approaches collect and analyze data of all users to predict what users like based on their similarity to other users. Content-based filtering (CBF) approach recommends items based on how much an item's description matches a user's preference profile, whereas Matrix Factorization (MF) characterizes both items and users by vectors of factors inferred from item rating patterns in which a high correspondence between item and user factors leads to a recommendation. Each

[*] This article is a revised and expanded version of a paper entitled "Recommending Books for Children Based on the Collaborative and Content-Based Filtering Approaches" presented at ICCSA'16, July 2016.

of these approaches has its strengths and its weaknesses—CF does not require understanding of the items for it to work while it does not perform well when access to other users' data are limited, CBF's performance is not affected by how much of the other users' data are available, but it requires a complex algorithm to analyzes the users' profile, whereas MF works very well with sparse matrices, since a single user might have rated only a small percentage of possible items, but a major challenge of using MF is computing the mapping of users and items to factor vectors. Furthermore, instead of focusing on the ratings assigned by the users provided to items, MF characterizes both items and users by vectors of factors inferred from item rating patterns to predict ratings [11].

These days one of the most commonly-recommended items is reading materials. Reading is an activity performed on a daily basis: from reading news articles and books to cereal boxes and street signs. We recognize that children literacy forms a foundation upon which children will gage their future reading.[1] It is imperative to motivate young readers to read by offering them appealing books to read so that they can enjoy reading and gradually establish a reading habit during their formative years that can aid in promoting their good reading habits. As stated in [31], learning to read is a key milestone for children living in a literate society, specially given that reading provides the foundation for children's academic success. A recent study [28] highlights the fact that children who "do not read proficiently by the end of third grade are four times more likely to leave school without a diploma than proficient readers." The results of the study correlate with earlier statistics [10] which confirm that 88% of children who are poor readers by the end of the first grade remain so by the end of the fourth grade. Moreover, young readers who successfully learn to read in the early primary years of school will more likely be prepared to read for pleasure and learning in the future [15]. The aforementioned findings constitute the essence of encouraging good reading habits early on. Identifying books appealing to children, however, can be challenging, given the amount of books made available on a regular basis that address a diversity of topics and target readers at different reading levels. It is essential to provide children with reading materials matching their preferences/interests and reading abilities, since exposing young readers to materials that are either too easy/difficult to understand or involving unappealing topics could diminish their interest in reading [1].

With the huge volume of children books available these days,[2] it is a time-consuming and tedious process for K-12[3] teachers, librarians, and parents to manually examine the topic of each book and choose the one for their students/children to read. Moreover, it is hard for children to choose books to read on their own, since they lack of experiences on choosing appropriate books to read. Online book websites, such as goodreads.com and commensensemedia.org, make book recommendations to children based on the popu-

[1] http://www.deafed.net/publisheddocs/sub/9807kle.htm

[2] According to a report published by the Statistics Portal (http://www.statista.com/statistics/194700/us-book-production-by-subject-since-2002-juveniles/), there are 32,624 children books published in the U.S.A. in 2012 alone.

[3] K-12, which is a term used in the educational system in the United States and Canada (among other countries), refers to the primary and secondary/high school years of public/private school grades prior to college. These grades are kindergarten (K) through 12^{th} grades.

larity and rankings. However, a child may not enjoy reading a popular book or a book with a very high ranking. For example, the book “Where the wild things are” is considered one of the most popular children books; however, some children find it unappealing to them because of the frightening scenes depicted in the book. Instead of solely relying on popularity or rankings on books, we have developed $CBRec$, a children book recommender, which adopts the *content-based* and *matrix factorization* approaches to make personalized book recommendations for children. The user-based and item-based collaborative filtering (CF) approaches are popular techniques for generating personalized recommendations [6]; however, when data sparsity becomes a problem for certain children users, i.e., when there is not enough data to generate similar user groups or similar item groups to use the CF methods, the content-based filtering and matrix factorization approaches [23] can be adopted to make personalized recommendations for users.

CBRec is designed for solving the *information overload* problem while minimizing the *time* and *efforts* imposed on parents/educators/young readers in discovering unknown, but suitable, books for pleasure reading or knowledge acquisition. CBRec first infers the readability level of a user U by analyzing the grade levels of books in his/her profile, which are determined using ReLAT, a robust readability level analysis tool that we have developed [21]. Hereafter, CBRec identifies a set of candidate books, among the ones archived at a website, with grade levels compatible to the inferred readability level of U. The current implementation of CBRec is tailored towards recommending books written in English and classified based on the K-12 grade level system. CBRec, however, can be easily adopted to make suggestions in languages other than English.

CBRec is a novel recommender that exclusively targets children readers, an audience who has not been catered by existing recommendation systems. CBRec is a self-reliant recommender which, unlike others, does not rely on personal tags nor access logs to make book recommendation. CBRec is unique, since it explicitly considers the ratings and content descriptions on books rated by children, in addition to the readability levels of children.

The remaining of this paper is organized as follows. In Section 2, we discuss existing book recommenders that have been used for suggesting books for individual readers, including children. In Sections 3, we introduce CBRec and its overall design methodology. In Section 4, we present the results of the empirical study on CBRec conducted to assess its performance. In Section 5, we give a concluding remark and present directions for future work on CBRec.

2 Related Work

In this section, we present a number of widely-used book recommenders and compare them with CBRec.

A number of book recommenders [14, 19, 34] have been proposed in the past. Amazon’s recommender [14] suggests books based on the purchase patterns of its users. Yang et al. [34] analyze users’ access logs to infer their preferences and apply the traditional CF strategy, along with a ranking method, to make book suggestions. Givon and Lavrenko [8] combine the CF strategy and social tags to capture the content of books for recommendation. Similar to the recommenders in [8, 34], the book recommender in [26] adopts the standard user-based CF framework and incorporates semantic knowledge in the form of a domain ontology to determine the users' topics of interest. The recommenders in [8, 26, 34] overcome the problem that arises due to the lack of initial information to perform the recommendation task, i.e., the cold-start problem. However, the authors of [8, 34] rely on user access logs and social tags, respectively to recommend books, which may not be publicly available and are not required by CBRec. Furthermore, the recommender in [26] is based on the existence of a book ontology, which can be labor-intensive and time-consuming to construct [7].

In making recommendations, Park and Chang [19] analyze individual/group behaviors, such as clicks and shopping habits, and features describing books, such as their library classification, whereas $PReF$ [20] suggests books bookmarked by connections of a LibraryThing user. $PReF$ adopts a similarity-matching strategy that uses analogous, but not necessarily the same, words to the ones employed to capture the content of a book of interest to U. This strategy differs from the exact-matching constraint imposed in [19] and a number of content-based recommenders [9, 17]. However, neither $PReF$ nor any of the aforementioned recommenders considers the readability level of their users as part of their recommendation strategies.

Vaz et al. [29] present a hybrid book recommendation system that take into account the preferences of users on the content of a book and its authors using two item-based CF approaches. Users' rankings on authors are predicted and are considered along with former book predictions for the users. Mooney and Roy [16], on the other hand, introduce a content-based book recommendation strategy which uses information about an item to make suggestions. An advantage of using the content-based approach for information filtering is that it does not rely on users' ratings on items which is useful in recommending previously unrated items. However, as shown in our empirical study (as presented in Section 4), the performance of a recommendation system based on either the item-based CF approach or content-based approach cannot achieve the same degree of effectiveness by incorporating the hybrid-based filtering approaches.

As mentioned earlier, Givon and Lavrenko [8] predict user ratings on books by using tags attached to books on a social-networking sites. The authors attempt to solve the cold-start book recommendation problem by inferring the most probable tags from the text of a book. However, there are major design faults of the proposed book recommendation system. First of all, According to [4], only 7.7% of published books in the OCLC database, a popular and worldwide library cooperative, are linked to the partial or full content of their corresponding books. For this reason, it is a severe constraint imposed on any analysis tool that relies on even an excerpt of a book due to copyright laws that often prohibit book content from being made publicly accessible. Second, tags are not widely available at children's book sites, since *personal tags* [13] assigned to books are rarely provided by children at the existing social bookmarking sites established for them.

Woodruff et al. [32] apply spreading activation over a text document (i.e., books in their case) and its citations such that nodes in the activation represent documents, whereas edges are created using the citations. The authors claim that the fused spreading activation techniques are superior compared with the traditional text-based retrieval methods. However, unlike textbooks or reference books in the book market, children books lack of references and hence the spreading activation methodology is not applicable to the design of children book recommendation systems.

Cui and Chen [5] claim that existing book recommendation systems do not offer enough information for their users to decide whether a book should be recommended to others. In solving the existing problem, the authors create recommendation pages of books which contain book information for the users to refer to. This recommendation approach, however, is not applicable to children, since the latter are interested in books recommended to them, instead of initiating the process of making recommendations on books to friends or other users on an online book websites.

3 Our Book Recommender

Content-based recommendation systems suggest books similar in *content* to the ones a given user has liked in the past, whereas recommendation systems based on the CF approaches identify a group of users S whose preferences represented by their ratings are similar to those of the given user U and suggests books to U that are likely appealing to U based on the ratings of S. Matrix factorization, on the other hand, characterizes both items and users by vectors of factors inferred from item rating patterns and yields a recommendation when there is a high correspondence between item and user factors.

3.1 Identifying Candidate Books

We recognize that "reading for understanding cannot take place unless the words in the text are accurately and efficiently decoded" [18] and only recommends books with readability levels appropriate to its users. To accomplish this task, CBRec determines the readability level of a book (user, respectively) using ReLAT [21] developed by us. Due to the huge number of books written for K-12 readers, it is not feasible to analyze all the books (e.g., books posted at a social bookmarking site) to identify the ones that potentially match the interests of a site user U. Consequently, CBRec follows a common practice among existing readability analysis tools [33] and applies Equation 1 to estimate the readability level of a user U, denoted $RL(U)$, based on the grade level of each book P_B in U's profile predicted by ReLAT, denoted ReLAT(P_B). Note that only books bookmarked in a user's profile during the most recent academic year are considered, since it is anticipated that the grade levels of books bookmarked by users gradually increase as the users enhance their reading comprehension skills over time.

$$RL(U) = \frac{\sum_{P_B \in P} ReLAT(P_B)}{|P|} \tag{1}$$

where $|P|$ denotes the number of books in U's profile and *average* is employed to capture the *central tendency* on the grade levels of books bookmarked by U.

CBRec first creates *CandBks*, the subset of books (archived at a social bookmarking site) that are compatible with the readability level of a user U which are further analyzed for making recommendations to U to ensure that recommendations made for U can be understood by and are suitable for U. *CandBks* includes a number of books considered

Table 1. A number of BiblioNasium books

ID	Book Title	Grade Level
Bk_1	Mummies in the Morning	2.9
Bk_2	Captain Underpants and the Big, Bad Battle of the Bionic Booger Boy	4.7
Bk_3	The Hidden Boy	5.6
Bk_4	Dragon's Halloween	3.1
Bk_5	Junie B. Jones Smells Something Fishy	3.0
...	...	...

by CBRec for recommendation, each of which is within-one-grade-level range from U's. By considering books within *one* grade level above/below U's mean readability level, [4] CBRec recommends books with an appropriate level of complexity for U and grade levels approximate to the grade levels of books that have been read by U (as of the most recent academic year) and thus encourage users' reading growth which are neither too difficult nor too easy for U to understand.

Example 1 Consider a user U who has bookmarked a number of books from Dav Pilkey's "Captain Underpants" series. Based on the grade levels predicted by ReLAT for the books archived at BiblioNasium.com (see a sample of BiblioNasium books in Table 1) and U's readability level, which is 4, CBRec does not include Bk_1 nor Bk_3 in $CandBks$, since their grade levels are below/beyond the range deemed appropriate for U and thus it is not considered for recommendation by CBRec for U. □

3.2 The Content-Based Filtering Method

The content-based filtering approach recommends items to a user that are *similar* to the ones that the user prefers in the past. The approach can be adopted for identifying the common characteristics of books being liked by user u and recommend to u new books that share these characteristics. The *similarity of books* is computed by using the designated features applicable to the books to be compared. For example, if u offers very high ratings on books in the domain of adventure or a particular author, then the content-based filtering approach suggests other books to u based on the same domain, i.e., adventure, or author.

The Content-Based Filtering Approach Using the Vector Space Model The content-based filtering method analyzes the descriptions of children books rated by a user u and construct the profile of u based on the descriptions which are used for predicting the ratings of books unknown to u. Given the attributes of a user profile that capture the preferences and interests of the corresponding user, a content-based recommender attempts to match the attributes with the ones that describe the content of another (new) book. This method does not require the *ratings* on books given by other users as in

[4] We have experimentally determined this range to ensure the suitability of the recommended books with respect to the reading level of the corresponding user.

the collaborative filtering approaches to predict ratings on unknown books to u. The user profile of u is a vector representation of u's interests, which is constructed using Equation 2.

$$X_u = \Sigma_{i \in \tau_u} r_{u,i} X_i \tag{2}$$

where τ_u is the set of books rated by user u, $r_{u,i}$ denotes the rating provided by user u on book i, and X_i is the vector representation of the description D on i with the *weight* of each keyword k in D computed by using the *term frequency (TF)* and *inverse document frequency (IDF)* of k.

The vector space model (VSM) is used to predict the *rating* of a book B unknown to user u using the profile P of u, denoted $BkSim(P, B)$. The profile representation of P is computed using Equation 2, whereas the vector representation of the description of B is determined similarly as X_i in Equation 2, i.e., the *weight* of each keyword k in the description of B, denoted B_i, is calculated by using the TF/IDF of k.

$$BkSim(P, B) = \frac{\Sigma_{i=1}^{t}(P_i \times B_i)}{\sqrt{\Sigma_{i=1}^{t} P_i^2 \times \Sigma_{i=1}^{t} B_i^2}} \tag{3}$$

where t is the dimension of the vector representation of P and B.

The Content-Based Filtering Approach Using the Naïve Bayes Model Besides using the vector space model for content-based filtering, machine-learning techniques have also been widely used in inducing content-based profiles. In using the machine-learning approach for text (which can be adopted for profile) classification, an inductive process automatically constructs a text classifier by learning from a set of training documents, which have already been labeled with the corresponding categories. Indeed, learning to classify user profiles can be treated as a binary text categorization problem, i.e., each item is classified as either interesting or not interesting with respect to the user preferences as specified by the attributes in a user profile. In this section, we discuss the Naïve Bayes classifier, which is a widely-used machine-learning algorithm in content-based filtering approach. Even though a constraint imposed on using the machine-learning approach for content-based filtering is that items must be labeled with their respective classes during the training process, after the classifier has been trained, it can be used to automatically infer profiles based on the trained model.

The Naïve Bayes model is a probabilistic approach to inductive learning. The probabilistic classifier developed by the Naïve Bayes model is based on the *Bayes' rule* as defined below.

$$\begin{aligned} P(C|D) &= \frac{P(D|C) \times P(C)}{P(D)} \\ &= \frac{P(D|C) \times P(C)}{\sum_{c \in C} P(D|C = c) P(C = c)} \end{aligned} \tag{4}$$

where C (D, respectively) is a random variable corresponding to a class (document[5], respectively.

Based on the term, which is either an attribute or a keyword in an item, independence assumption, the Naïve Bayes rule yields

$$\begin{aligned} P(c|d) &= \frac{P(d|c) \times P(c)}{\sum_{c \in C} P(d|c)P(c)} \\ &= \frac{\prod_{i=1}^{n} P(w_i|c) \times P(c)}{\sum_{c \in C} \prod_{i=1}^{n} P(w_i|c) \times P(c)} \end{aligned} \tag{5}$$

where w_i ($1 \leq i \leq n$) is a term in d, and $\sum_{c \in C} \prod_{i=1}^{n} P(w_i|c) \times P(c)$ is a *chain rule*.

The classification process is based on the following computation:

$$\begin{aligned} Class(d) &= arg\ max_{c \in C} P(c|d) \\ &= arg\ max_{c \in C} \frac{P(d|c) \times P(c)}{\sum_{c \in C} P(d|c) \times P(c)} \end{aligned} \tag{6}$$

where $P(d|c)$ is the *probability* that d is observed, given that the class is known to be c, and $P(c)$ is the *probability* of observing class c, which is defined as

$$P(c) = \frac{N_c}{N} \tag{7}$$

where N_c is the number of training items in class c, and N is the total number of training items.

In estimating $P(d|c)$ in Equation 6, the *Multiple-Bernoulli model* is applied, since the Multiple-Bernoulli distribution is a natural way to model the distributions over binary vectors. $P(d|c)$ is computed in the *Multiple-Bernoulli model* as

$$P(d|c) = \prod_{w \in V} P(w|c)^{\delta(w,d)} (1 - P(w|c))^{1 - \delta(w,d)} \tag{8}$$

where $\delta(w, d)$ = 1 if and only if term w occurs in d.

Note that $P(d|c)$ = 0 if there exists a $w \in d$ that never occurs in c in the training set, which is the *data sparseness* problem and can be solved by using the *Laplacian smoothed* estimate as defined below.

$$P(w|c) = \frac{df_{w,c} + 1}{N_c + 1} \tag{9}$$

where $df_{w,c}$ denotes the number of items in c which includes term w, and N_c is the number of items belonged to the class c.

In designing CBRec, we have decided to adopt the *vector space model* instead of the *Naïve Bayes* model, since the latter requires a trained model using a pre-defined

[5] From now on, unless stated otherwise, a document is treated as an item, such as a book.

labeled item set which imposes additional overhead. However, the *Naïve Bayes* model is an alternative model that can be considered in developing CBRec or other book recommender systems for children.

3.3 The Collaborative Filtering (CF) Approaches

The CF approaches rely on the ratings of a user and other users. The predicted rating of a user u on a book i is likely similar to the rating of user v on i if both users have rated other books similarly.

The User-Based CF Approach The user-based CF approach determines the interest of a user u on a book i using the ratings on i by other users, i.e., the neighbors of u who have similar rating patterns [36]. We apply the Cosine similarity measure as defined in Equation 10 to calculate the similarity between two users u and v and determine user pairs which have the lowest rating difference among all the users. Equation 10 can be applied to compute the *similarity* between a user and each one of the other users to find out the *similarity group* of each user. Two users who have the lowest difference rating value between them means that they are the *closest neighbors*.

$$USim(u,v) = \frac{\varSigma_{s \in S_{u,v}}(r_{u,s} \times r_{v,s})}{\sqrt{\varSigma_{s \in S_{u,v}} r_{u,s}^2} \times \sqrt{\varSigma_{s \in S_{u,v}} r_{v,s}^2}} \tag{10}$$

where $r_{u,s}$ ($r_{v,s}$, respectively) denotes the rating of user u (user v, respectively) on book s, and $S_{u,v}$ denotes the set of books rated by both users u and v.

Upon determining the K-nearest neighbors (i.e., KNN) of a user u using Equation 10, we can compute the predicted rating on a book s for u using Equation 11.

$$\hat{r}_{u,s} = \bar{r_u} + \frac{\varSigma_{v \in S_{u,v}}(r_{v,s} - \bar{r_u}) \times USim(u,v)}{\varSigma_{v \in S_{u,v}} USim(u,v)} \tag{11}$$

where $\hat{r}_{u,s}$ stands for the predicted rating on book s for user u, $\bar{r_u}$ is the average rating on books provided by user u, $S_{u,v}$ is the group of closest neighbors of u, $r_{v,s}$ is the rating of user v on book s, and $USim(u,v)$ is the similarity measure between users u and v as computed in Equation 10.

Instead of using the user-based predicted ratings as defined in Equation 11, another commonly-used user-based rating prediction approach is given in Equation 12 in which the rating prediction is computed for each user u on a new book i without considering *different levels of similarity* among users.

$$\hat{r}_{u,i} = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{v,i} \tag{12}$$

where $N_i(u)$ is the group of nearest neighbors of u who have rated book i and $r_{v,i}$ is the rating of i provided by user v who is one of the nearest neighbors of u.

If the nearest neighbors of u come with different levels of similarity with respect to u, denoted $w_{u,v}$, the predicted user-based rating using the different levels of user similarity is defined as

Table 2. Children's books and the ratings (in the range of 1-5) by the children

	Runny Babbit	Harry Potter	Kid Athletes	Funny Bones	Finding Winnie
Noah		4	2	3	3
Alex	5	4	3	1	4
Emma	3	4	?	1	5
Ava	4	3	4	2	5
Jacob	2			5	2

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_i(u)} w_{u,v} \times r_{v,i}}{\sum_{v \in N_i(u)} w_{u,v}} \tag{13}$$

Example 2 Consider Table 2 which includes a number of children and their ratings on different books.

Two children, Emma and Ava, have very similar ratings on the five books listed in Table 2, whereas Emma and Jacob have very dissimilar ratings on corresponding books. Both Emma and Ava enjoyed the book *Finding Winnie* and disliked the book *Funny Bones*. However, Jacob really liked the book *Funny Bones*, whereas Emma did not like it at all.

Assume that we are supposed to predict the rating of the book *Kid Athletes* for Emma using the ratings provided by Ava and Alex, the two nearest neighbors of Emma. Further assume that the similarity values between Emma and Ava and between Emma and Alex are 0.8 and 0.5. Applying Equation 13, the predicted rating on the book *Kid Athletes* for Emma is

$$\hat{r}_{\text{``}Emma\text{''},\text{``}KidAthletes\text{''}} = \frac{0.8 \times 4 + 0.5 \times 3}{0.8 + 0.5} \cong 3.62 \ \square$$

We adopt Equation 11 for the rating predictions using the user-based CF approach, instead of Equation 13, which requires different levels of similarity among different users to be generated in advance. However, if the similarity weights are available among different users, Equation 13 could be adopted in place of Equation 11 in the user-based CF rating prediction.

The Item-Based CF Approach Contrast to the user-based CF approach which relies on similar user groups to recommend books, the item-based CF approach computes the similarity values among different books and determines sets of books with similar ratings provided by different users. The item-based CF approach predicts the rating of a book i for a user u based on the ratings of u on books similar to i. The *adjusted cosine similarity matrix* [30] is applied to compute the similarity values among different books and assign books with similar ratings into the same similar-item group as defined in Equation 14.

$$ISim(i,j) = \frac{\Sigma_{u \in U}(r_{u,i} - \bar{r_u}) \times (r_{u,j} - \bar{r_u})}{\sqrt{\Sigma_{u \in U}(r_{u,i} - \bar{r_u})^2} \times \sqrt{\Sigma_{u \in U}(r_{u,j} - \bar{r_u})^2}} \tag{14}$$

where $ISim(i, j)$ denotes the similarity value between books i and j, $r_{u,i}$ ($r_{u,j}$, respectively) denotes the rating of user u on book i (j, respectively), $\bar{r_u}$ is the average rating for user u on all books u has rated, and U is the set of books rated by u.

Equation 14 computes the similarity between two books, whereas Equation 15 predicts the rating for user u on book i.

$$S_{u,i} = \bar{r_u} + \frac{\Sigma_u (r_{u,i} - \bar{r_u})}{u(i) + r} + \frac{\Sigma_{j \in I(u)} (r_{u,j} - \frac{\Sigma_u (r_{u,j} - \bar{r_u})}{u(j)})}{I(u) + r} \tag{15}$$

where $S_{u,i}$ denotes the predicted rating on book i for user u, $\bar{r_u}$ denotes the *average rating* on all books u has rated, $r_{u,i}$ ($r_{u,j}$, respectively) is the rating on book i (j, respectively) provided by u, $I(u)$ is the set of books u has rated, $u(j)$ is the number of users who has rated i, and r is the book rating which is used to decrease the extremeness when there are not enough ratings available, which is determined experimentally.

The first component on the right-hand side of Equation 15 is called the *global mean* which is the average rating on all the books u has rated. The second component is called the *item offset* which is the score for user u on book i, whereas the third component is called the *user offset* which is the user prediction on book i.

An alternative item-based CF approach is presented in [6] which considers similar items (i.e., books in our case) with similarity weights between two items, which are predefined. The predicted rating on item i for user u is computed as follows.

$$\hat{r}_{u,i} = \frac{\sum_{j \in N_u(i)} w_{i,j} \times r_{u,j}}{\sum_{j \in N_u(i)} |w_{i,j}|} \tag{16}$$

where $N_u(i)$ is the set of items rated by user u that are *most similar* to item i, and $w_{i,j}$ is the *similarity weight* between items i and j.

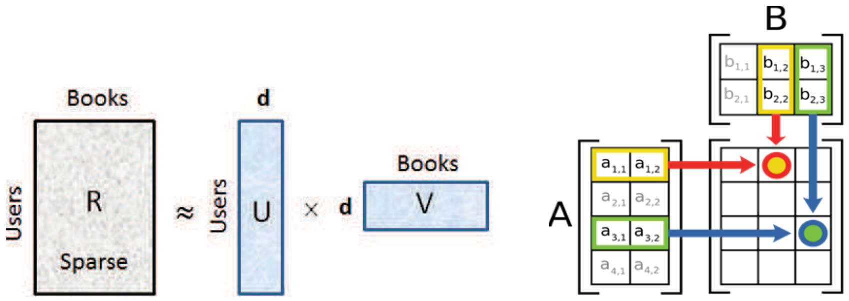
Example 3 Consider Example 2 again. Instead of consulting Emma's peers, CBRec considers the ratings on the books Emma and others have read in the past. Based on the ratings provided by children as shown in Table 2, the two books that are the closest neighbors, i.e., most similar in terms of ratings, of the book *Kid Athletes* are *Harry Potter* and *Finding Winnie*. Assume that the similarity values between the books *Kid Athletes* and *Harry Potter* and between *Kid Athletes* and *Finding Winnie* are 0.55 and 0.35. As shown in Table 2, the ratings given by Emma on *Harry Potter* and *Finding Winnie* are 4 and 5, respectively, the predicted rating on *Kid Athletes* for Emma is

$$\hat{r}_{\text{``}Emma\text{''},\text{``}KidAthletes\text{''}} = \frac{0.55 \times 4 + 0.35 \times 5}{0.55 + 0.35} \cong 4.3 \quad \square$$

We adopt Equation 16 for our item-based CF approach, since the similarity weights of two items can easily be predefined using contents on items and Equation 16 is widely-used in item-based Collaborative Filtering approaches.

3.4 The Matrix Factorization (MF) Approach

Matrix Factorization is getting more popular in the recent years. Although user-based and item-based filtering perform recommendation well, Matrix Factorization outper-



(a) Visual representation of Matrix Factorization

(b) Matrix multiplication

Fig. 1. The Matrix Factorization Model

forms these models [11, 23]. One main reason behind it is that user-based and item-based offer recommendation based on similarity. MF is a good approach, especially whenever the dataset is sparse, since the user-based and item-based filtering approaches are forced to recommend using neighbors that are not really that similar. Moreover, Matrix Factorization does not require similar users or items to give recommendation. Thus, it is more stable than the other recommendation models.

Latent Factor Model Although Matrix Factorization also relies on the user and item ratings, it does not adopt the neighborhood methods to find similar items or users. Instead, it considers different *latent features* to train a model to predict users' ratings on items. Latent factors are inferred variables that contribute to how a user rates an item. In the context of a book, latent factors can be the amount of fiction in a book or how popular the main character is. The goal of Matrix Factorization is to infer these latent factors from the given users' ratings matrix. Once the latent factors are extracted from the matrix, each user and item is mapped onto a latent feature space [24]. These user and item mappings are represented as a feature vector. Feature vector is the amount of association between the feature and the user, or the item. For example, a user feature vector can be to what degree a user like fiction in a book, whereas an item feature vector can be the portion of fiction in a book. The Matrix Factorization model recommends items that are close to the user feature vector in the latent feature space.

Matrix Factorization Model Matrix Factorization, as its name reflects, factors a matrix into different matrices. As shown in Figure 1(a), R is a $n \times m$ matrix, where n represents the number of users and m denotes the number of items, and $R[i][j]$ is the rating of the i^{th} user on the j^{th} item $(1 \leq i \leq n, 1 \leq j \leq m)$. The two new matrices are U and V of size $n \times d$ and $d \times m$, respectively, where d is the number of latent features. The row in matrix U represents the *magnitude* of the user feature vector, i.e., the amount of association between the user and the latent features, whereas the column in

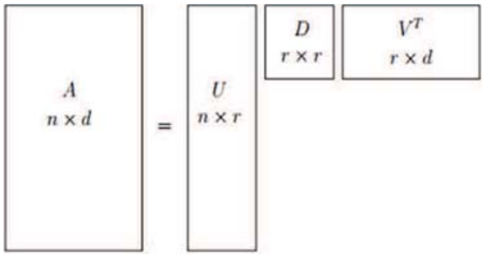


Fig. 2. The SVD decomposition of an $n \times d$ matrix

matrix V represents the amount of *association* between an item and the latent features. The predicted rating of the i^{th} user on the j^{th} item can be calculated by multiplying the i^{th} row in U with the j^{th} column in V together (see Figure 1(b)). Multiplying the matrices U and V together yields the matrix R', which now also includes the predicted ratings that were previously unknown which are calculated based on the d latent factors.

Among the proposed Matrix Factorization algorithms, there are a number of different models to factorize a matrix. Some of the commonly-used techniques include singular value decomposition (SVD), principal component analysis (PCA), and non-negative matrix factorization(NMF).

- **Singular Value Decomposition (SVD)** is a dimensionality reduction approach. SVD of an $n \times d$ matrix A is of the form UDV^T, where U and V^T are $n \times r$ and $r \times d$ orthogonal matrices, respectively, and D is the $r \times r$ singular orthogonal matrix with non-negative elements (see Figure 2).

  The diagonal elements in D, i.e., σ_1, σ_2, $\ldots$, σ_n, are *singular values* of matrix A. Singular values are square roots of the eigenvalues of the of $A^T \times A$, where A^T is the conjugate transpose of A, and usually, the singular values are placed in the descending order in D. The data matrix A is useful for finding a low-rank matrix, which is a completed matrix with *predicted ratings*, which can be used to predict the missing values. The column vectors of U and V^T are *left singular vectors* and *right singular vectors*, respectively [2]. Using the singular values, the left singular vectors, and the right singular vectors, we can create the matrix A, which contains the predicted ratings of missing values in A.

- **Principal Component Analysis (PCA)** is a statistical method to find patterns in high dimensional data sets. PCA obtains an ordered list of components that account for the largest amount of the variance from the data in terms of least square errors. In other words, the components contribute to the reason why users rate one item higher than the other. The components are the underlying structure in the data. They are the directions where there is the most variance, the directions where the data is most spread out. The amount of variance captured by the first component is larger than the amount of variance on the second component and so on. We can reduce the dimensionality of the data by neglecting those components that gives small variance to the data, since they do not affect how the users rate items [3].

While SVD is a numerical method to factorize a matrix, which reconstructs the original matrix based on the matrices, PCA is a statistical method that focuses on summarizing data. Once the data is summarized, PCA attempts to reconstruct the original data using the summarized data. Even though both SVD and PCA approaches have the same design goal, they account for different aspects of the data to obtain the reconstructed matrix.

- **Non-Negative Matrix Factorization (NMF)**. Given a non-negative matrix R, we are supposed to find non-negative matrix factors U and V such that $R \approx UV$. One way to approach this problem is to first initialize the two matrices with some values, calculate how "different" their product is to V, and then try to minimize this difference iteratively.

  NMF can be applied to the statistical analysis of multivariate data in the following manner. Given a set of multivariate n-dimensional data vectors, the vectors are placed in the columns of an $n \times m$ matrix R. This matrix is then approximately factorized into an $n \times d$ matrix U and an $d \times m$ matrix V. Usually, d is chosen to be smaller than n or m, so that U and V are smaller than the original matrix R. This results in a compressed version of the original data matrix [12] (See Figure 1(a)).

Even though there are many different models to factorize matrix, SVD is the most popular one among all of them. SVD is capable of handling *massive* dataset, *sparseness* of rating matrix, and the *cold-start* problem, i.e., not having enough data to predict rating accurately initially. Even though SVD requires complicated calculation and multiple adjustment of the algorithm to produce good prediction, it produces the most accurate prediction over other MF algorithms. For this reason, CBRec adopts the SVD technique in Matrix Factorization.

4 Experimental Results

In this section, we first introduce the datasets used for the empirical study conducted to assess the performance CBRec (in Section 4.1). Hereafter, we present the results of the empirical study on CBRec in Section 4.2 and compare the performance of CBRec with current state-of-the-art book recommendation systems in Section 4.3.

4.1 Datasets

We have chosen a number of children book records included in the Book-Crossing dataset to conduct our performance evaluation of CBRec.[6] The book-crossing dataset was collected by Cai-Nicolas Ziegler [37] between August and September of 2004 with data extracted from BookCrossing.com. It includes 278,858 users who provide, on the scale of 1 to 10, 1,149,780 ratings on 271,379 books. Since not all of books in the

[6] Other datasets can be considered as long as they contain user_IDs, book ISBNs, and rating information.

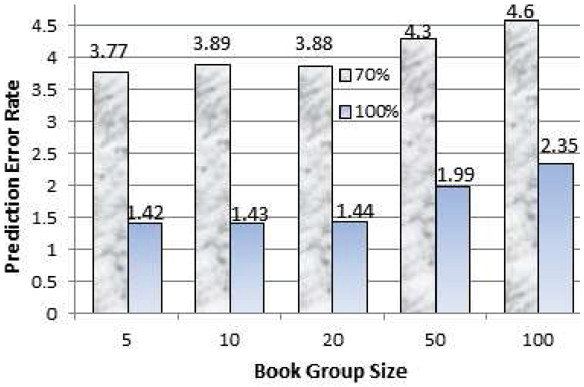


Fig. 3. Prediction errors of the content-based approach on the children books in the Book-Crossing dataset

book-crossing database are children books, we pre-processed the dataset to extract only children book records. Each record includes a user_ID, the ISBN of a book, and the rating provided by the user (identified by the user_id) on the book. We used Amazon.com AWS advisement API to verify that the ISBNs from the book-crossing dataset are valid and they are children books. Out of the 271,379 books in the Book-Crossing dataset, approximately 29,000 books are children books, which is denoted as CBC_DS.

Besides using the children books and their ratings in the Book-Crossing dataset, we extracted the book description of 30% of the children books in CBC_DS from Amazon.com, since they were missing in the children books and needed for the content-based filtering approach. The Amazon dataset yields the additional dataset used for evaluating the performance of CBRec. Figure 3 shows the differences in terms of prediction errors using only 70% versus 100% of book descriptions generated by the content-based filtering approach.

4.2 Performance Evaluation of CBRec

In our empirical study, we considered the similar user group size of *ten* users in the user-based CF approach, since LensKit,[7] which has implemented the user-based CF method and has been cited in a number of published papers, has demonstrated that ten is an ideal group size in predicting user ratings. We have also chosen *ten* to be the group size of books used in the content-based and item-based CF approaches, since the prediction error rate using this group side is the most ideal, in terms of size and accuracy, as demonstrated in our empirical study and reported in Figure 3.

To evaluate the performance of CBRec, which is based on the implementation of the item-based and user-based CF, content-based, and MF approaches in LensKit, we computed the prediction error of CBRec for each user U in CBC_DS by taking the absolute value of the difference between the real and predicted ratings on each book

[7] http://lenskit.org/

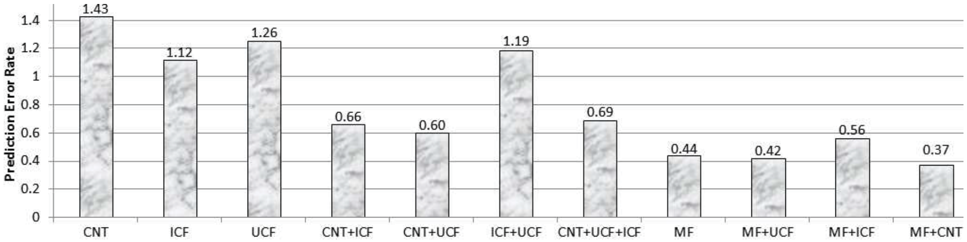


Fig. 4. Prediction error rates of the various filtering approaches, CNT (Content), ICF (Item-based CF), UCF (User-based CF), MF (Matrix Factorization), and their combinations

U has rated in the dataset. These prediction errors were added up and divided it by the total number of predictions. Figure 4 shows the prediction error rate of each filtering approaches and their combinations.

As shown in Figure 4, the combined Matrix Factorization (MF) and content-based (CNT) approach, which is adopted by CBRec, outperforms individual and other combined prediction models in terms of obtaining the lowest prediction error rates among all the models. CBRec achieves the highest prediction accuracy, which is less than *half* a rating (out of 10) away from the actual rating, since the CNT approach compensates the MF approach when user content is available. The prediction error rate, i.e., accuracy ratio, achieved by CBRec is statistically significant ($p < 0.05$) over the individual MF, content-based, and item-based CF, and user-based CF, approach as well as their combinations. The experimental results have verified that CBRec is the most accurate recommendation tool in predicting children's ratings on books, which is the most suitable choice for making book recommendations for children based on the rating prediction.

4.3 Comparing CBRec with Other Book Recommendation Systems

In this section, we detail the book recommenders to be compared with CBRec. These recommenders were chosen, since they achieve high accuracy in recommendations on books based on their respective model.

– **MF**. Yu et al. [35] and Singh et al. [27] predict ratings on books and movies based on matrix factorization (MF), which can be adopted for solving large-scale collaborative filtering problems. Yu et al. develop a non-parametric matrix factorization (NPMF) method, which exploits data sparsity effectively and achieves predicted rankings on items comparable to or even superior than the performance of the state-of-the-art low-rank matrix factorization methods. Singh et al. introduce a collective matrix factorization (CMF) approach based on relational learning, which predicts user ratings on items based on the items' genres and role players, which are treated as unknown values of a relation between entities of a certain item using a given database of entities and observed relations among entities. Singh et al. propose different stochastic optimization methods to handle and work efficiently on large and sparse data sets with relational schemes. They have demonstrated that their model is practical to process relational domains with hundreds of thousands of entities.

- **ML**. Besides the matrix factorization methods, probabilistic frameworks have been introduced for rating predictions. Shi et al. [25] propose a joint matrix factorization model for making context-aware item recommendations.[8] Similar to CBRec, the matrix factorization model developed by Shi et al. relies not only factorizing the user-item rating matrix but also considers contextual information of items. The model is capable of learning from user-item matrix, as in conventional collaborative filtering model, and simultaneously uses contextual information during the recommendation process. However, a significant difference between Shi et al.'s matrix factorization model and CBRec is that the contextual information of the former is based on movie mood, whereas CBRec makes recommendations according to the contextual information on books.
- MudRecS [22], which makes recommendations on books, movies, music, and paintings similar in content to other books, movies, music, and/or paintings that a MudRecS user is interested in. MudRecS does not rely on users' access patterns/histories, connection information extracted from social networking sites, collaborated filtering methods, or user personal attributes (such as gender and age) to perform the recommendation task. It simply considers the users' ratings, genres, role players (authors or artists), and reviews of different multimedia items. MudRecS predicts the *ratings* of multimedia items that match the interests of a user to make recommendations.

Figure 5 shows the Mean Absolute Error and RMSE scores of CBRec and other recommendation systems on the CBC_DS dataset. *Root Mean Square Error* (RMSE) and *Mean Absolute Error* (MAE) are two performance metrics widely-used for evaluating rating predictions on multimedia data. Both RMSE and MAE measure the *average magnitude* of *error*, i.e., the average prediction error, on incorrectly assigned ratings. The error values computed by RMSE are squared before they are summed and averaged, which yield a relatively *high* weight to errors of *large* magnitude, whereas MAE is a *linear* score, i.e., the absolute values of individual differences in incorrect assignments are weighted equally in the average.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(f(x_i) - y_i)^2}{n}}, \qquad MAE = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i| \tag{17}$$

where n is the total number of items with ratings to be evaluated, $f(x_i)$ is the rating predicted by a system on item x_i ($1 \leq i \leq n$), and y_i is an expert-assigned rating to x_i.

As the MAE and RMSE scores shown in Figure 5, CBRec significantly outperforms other book recommendation systems on rating predictions of the respective books based on the Wilcoxon Signed-Ranks Test ($p \leq 0.05$).

5 Conclusions and Future Work

Existing book recommenders either are (i) not personalized enough, since they make the *same* recommendations to all users on a given book, (ii) based on the availability

[8] The system was originally designed to predict ratings on *movies* but was implemented by [22] for additional comparisons on *books* as well.

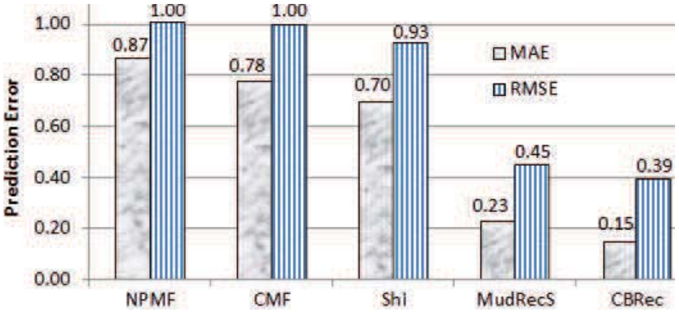


Fig. 5. The MAE and RMSE scores for various book recommendation systems based on CBC_DS, the children books of the BookCrossing, dataset

of users' historical data in the form of social tags, which may not be publicly available on children, or (iii) developed for a *general audience*, instead of taking into account the reading levels of their users. To address these issues, we have developed CBRec, a book recommender tailored to children, which simultaneously considers the reading levels and interests of its users in making personalized suggestions. CBRec adopts the widely-used *Content-based Filtering* approach and the *Matrix Factorization* approach and integrates the two filtering approaches in predicting ratings on children books to make book recommendation. Predicted ratings on children books, in addition to the readability levels of the (candidate) books to be considered for recommendation, provide CBRec the wealth of useful information to suggest books with appropriate levels of complexity and topics of interest that are appealing to children.

CBRec is unique, since it makes *personalized* suggestions on books that satisfy both the *preferences* and *reading abilities* of its users. Unlike current state-of-the-art recommenders that rely on the existence of user access logs or social tags, CBRec simply considers *brief descriptions* on children's books, their *ratings*, and their *grade levels* computed using our grade-level prediction tool, ReLAT, which is different from popular readability formulas that focus solely on analyzing lexicographical and syntactical structures of the texts in books. Information, such as metadata and ratings on books, are readily available on (children) social bookmarking websites, such as goodreads.com., and ReLAT can determine the grade level of any book (even if a sample of the text of a book is unavailable) by analyzing the Subject Headings of books, US Curriculum subject areas identified in books, and information about the authors of books. As children continue to read more books if they can *choose* what to read [1], a significant contribution of CBRec is to provide children a selection of suitable books to choose from that are not only appealing to them, but can be comprehended by them. The conducted experiments demonstrate the effectiveness of CBRec in suggesting books for children.

As a by-product of this research work we have created a benchmark dataset consisting of users and books, in addition to metadata and readability levels of the books, which can be used to assess the performance of recommenders that provide books suggestions to K-12 readers.

As part of our future work, we intend to extend CBRec so that it can suggest reading materials for struggling readers, especially the ones with learning disabilities and those who learn English as a second language. For these readers, their readability levels can be different from ordinary students. Book recommenders can aid these users by finding books potentially of interest to them.

References

1. R. Allington and E. Gabriel. Every Child, Every Day. *Reading: The Core Skill*, 69(6):10–15, 2012.
2. D. Bokde, S. Girase, and D. Mukhopadhyay. Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey. *International Journal of Advance Foundation and Research in Computer (IJAFRC)*, 1(12):111–118, December 2014.
3. D. Bokde, S. Girase, and D. Mukhopadhyay. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. *Procedia Computer Science*, 49:136–146, 2015.
4. X. Chen. Google Books and WorldCat: A Comparison of Their Content. *Online Information Review*, 36(4):507–516, 2012.
5. B. Cui and X. Chen. An Online Book Recommendation System Based on Web Service. In *Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'09)*, pages 520 – 524, 2009.
6. C. Desrosiers and G. Karypis. *Recommender Systems Handbook*, chapter A Comprehensive Survey of Neighborhood-based Recommendation Methods, pages 107–144. Springer, 2011.
7. Z. Ding. The Development of Ontology Information System Based on Bayesian Network and Learning. *Advances in Intelligent and Soft Computing*, 129:401–406, 2012.
8. S. Givon and V. Lavrenko. Predicting Social-Tags for Cold Start Book Recommendations. In *Proceedings of the 3rd ACM Conference on Recommender Systems (ACM RecSys 2009)*, pages 333–336, 2009.
9. Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He. Document Recommendation in Social Tagging Services. In *Proceedings of the 19th International World Wide Web Conference (WWW 2010)*, pages 391–400, 2010.
10. C. Juel. Learning to Read and Write: A Longitudinal Study of Fifty-Four Children from First Through Fourth Grade. *Educational Psychology*, 80:437–447, 1988.
11. Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, 42(8):30–37, 2009.
12. D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of Advances in Neural Information Processing Systems, Volume 13*, pages 556–562, 2001.
13. H. Li, Y. Gu, and S. Koul. Review of Digital Library Book Recommendation Models. SSRN (dx.doi.org/10.2139/ssrn.1513415), 2009.
14. G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.
15. Ministry of Education of Ontario. A Guide to Effective Instruction in Reading, Kindergarten to Grade 3. Available at http://goo.gl/UCo5e3, 2005.
16. R. Mooney and L. Roy. Content-Based BookRecommending Using Learning for Text Categorization. In *Proceedings of the fifth ACM conference on Digital libraries (DL'00)*, pages 195–204, 2000.
17. C. Nascimento, A. Laender, A. da Silva, and M. Goncalves. A Source Independent Framework for Research Paper Recommendation. In *Proceedings of ACM/IEEE Joint Conference on Digital Libraries (JCDL 2011)*, pages 297–306, 2011.

18. J. Oakhill and K. Cain. The Precursors of Reading Ability in Young Readers: Evidence from a Four-Year Longitudinal Study. *Scientific Studies of Reading (SSR)*, 16(2):91–121, 2012.
19. Y. Park and K. Chang. Individual and Group Behavior-based Customer Profile Model for Personalized Product Recommendation. *Expert Systems with Applications*, 36(2):1932–1939, 2009.
20. M.S. Pera and Y.-K. Ng. With a Little Help From My Friends: Generating Personalized Book Recommendations Using Data Extracted from a Social Website. In *Proceedings of the 2011 IEEE/WIC/ACM Joint Conference on Web Intelligent (WI'11)*, pages 96–99, 2011.
21. M.S. Pera and Y.-K. Ng. What to Read Next?: Making Personalized Book Recommendations for K-12 Users. In *Proceedings of the 7th ACM Conference on Recommender Systems (ACM RecSys 2013)*, pages 113–120, 2013.
22. R. Qumsiyeh and Y.-K. Ng. Predicting the Ratings of Multimedia Items for Making Personalized Recommendations. In *Proceedings of the 35th International ACM Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 475–484, 2012.
23. F. Ricci, L. Rokach, B. Shapira, and P. Kantor. *Recommender Systems Handbook*. Springer, 2011.
24. S. Schelter, C. Boden, and V. Markl. Latent Factor Models for Collaborative Filtering. https://www.slideshare.net/sscdotopen/latent-factor-models-for-collaborative-filtering, 2012.
25. Y. Shi, M. Larson, and A. Hanjalic. Mining Mood-Specific Movie Similarity with Matrix Factorization for Context-Aware Recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pages 34–40, 2010.
26. A. Sieg, B. Mobasher, and R. Burke. Improving the Effectiveness of Collaborative Recommendation with Ontology-based User Profiles. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems (ACM HetRec)*, pages 39–46, 2010.
27. A. Singh and G. Gordon. Relational Learning via Collective Matrix Factorization. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 650–658, 2008.
28. The Annie E. Casey Foundation. Early Warning Confirmed: A Research Update on Third-Grade Reading. Available at http://goo.gl/HQrPOA, 2013.
29. P. Vaz, D. de Matos, B. Martins, and P. Calado. Improving a Hybrid Literary Book Recommendation System Through Author Ranking. In *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'12)*, pages 387–388, 2012.
30. J. Wang, A. de Vries, and M. Reinders. Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*, pages 501–508, 2006.
31. G. Whitehurst and C. Lonigan. *Handbook of Early Literacy Research, Volume 1*, chapter Emergent Literacy: Development from Prereaders to Readers. The Guilford Press, 2003.
32. A. Woodruff, R. Gossweiler, J. Pitkow, E. Chi, and S. Card. Enhancing a Digital Book with a Reading Recommender. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI'00)*, pages 153–160, 2000.
33. B. Wright and A. Stenner. Readability and Reading Ability. ERIC Document Reproduction Service No. ED435979, 1998.
34. C. Yang, B. Wei, J. Wu, Y. Zhang, and L. Zhang. CARES: A Ranking-oriented CADAL Recommender System. In *Proceedings of ACM/IEEE Joint Conference on Digital Libraries (JCDL 2009)*, pages 203–212, 2009.
35. K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast Nonparametric Matrix Factorization for Large-Scale Collaborative Filtering. In *Proceedings of International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 211–218, 2009.

36. Z. Zhao and M. Shang. User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM KDD)*, pages 478–481, 2010.
37. C. Ziegler, S. McNee, J. Konstan, and G. Lausen. Improving Recommendation Lists Through Topic Diversification. In *WWW*, pages 22–32, 2005.