

Improving an Hybrid Literary Book Recommendation System through Author Ranking

Paula Cristina Vaz
INESC-ID/IST
Rual Alves Redol, 9
1000029 Lisbon, Portugal
paula.vaz@inesc-id.pt

Bruno Martins
IST/INESC-ID
Avenida Professor Cavaco Silva
2780990 Porto Salvo, Portugal
bruno.martins@ist.utl.pt

David Martins de Matos
INESC-ID/IST
Rual Alves Redol, 9
1000029 Lisbon, Portugal
david.matos@inesc-id.ptcm/cmr6.pt

Pavel Calado
IST/INESC-ID
Avenida Professor Cavaco Silva
2780990 Porto Salvo, Portugal
pavel.calado@ist.utl.pt

ABSTRACT

Literary reading is an important activity for individuals and choosing to read a book can be a long time commitment, making book choice an important task for book lovers and public library users. In this paper we present an hybrid recommendation system to help readers decide which book to read next. We study book and author recommendation in an hybrid recommendation setting and test our approach in the LitRec data set. Our hybrid book recommendation approach purposed combines two item-based collaborative filtering algorithms to predict books and authors that the user will like. Author predictions are expanded in to a book list that is subsequently aggregated with the former list generated through the initial collaborative recommender. Finally, the resulting book list is used to yield the top-n book recommendations. By means of various experiments, we demonstrate that author recommendation can improve overall book recommendation.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information filtering; H.3.7 [Digital Libraries]: System issues

General Terms

Algorithms, Experimentation

Keywords

Hybrid Recommender, Book Recommendation, Author Recommendation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

1. INTRODUCTION

Literary reading is an important activity for individuals. Public libraries make it possible to exercise this activity for free, by letting users borrow books for one or two weeks. In this context, choosing the right book to borrow becomes an important task, because it can save the reader unnecessary trips to the library to pick new books.

On-line recommendation systems have proved to be very useful helping, users through the suggestion of items that satisfy user needs or preferences. Good recommendations in a public library could improve reader's usability of the library. Libraries have limited shelf space, but still have enough books to make book selection difficult and time consuming. However, the number of books and users is not enough to successfully use the traditional collaborative techniques that rely on large amounts of data to detect patterns. Public library users have a limited number of books that can be borrowed each time they visit the library. In Portugal, most libraries set a limit of five books for a two week period. In this context, it is important that the top five recommendations have books preferred by the user.

This research work aims to (i) assess whether item-based collaborative filtering (ICF) can be used to make good recommendations in an a public library, and (ii) assess whether selecting books by author preferences can improve recommendations (a survey posted on *Goodreads*¹ revealed that 78% of the respondents choose the next book to read with basis on authorship).

In this paper, we purpose a weighted hybrid approach to recommend literary books, that can be used in the context of public libraries. Our approach combines two ICF algorithms to improve recommendations, where one recommends books (ICFB) and the other recommends authors (ICFA). Authors are used to improve the book top-n recommendations through a fusion approach.

2. BACKGROUND

In this section we briefly explain and classify recommendation systems. We also describe the score aggregation func-

¹<http://www.goodreads.com> is a social network for book readers.

tions used in our approach.

2.1 Collaborative filtering

Collaborative filtering is a technique used in recommendation where information is filtered using multiple sources. Literature on recommendation systems [1] distinguishes between two main types of collaborative filters, namely user-based (UCF) and item-based (ICF).

UCF tries to find like-minded users to produce recommendations. Generally, this type of algorithms have poorer performance than ICF algorithms [7].

ICF, popularized by *Amazon.com*, searches for commonalities between items to make recommendations. Traditional ICF systems represent items as an N-dimensional vector of users, where N is the number of users in the system. Each position of the vector contains the rating given by the user to the item. The algorithm computes an item similarity matrix, using an appropriate similarity function. The most common function used in these algorithms is the cosine similarity [7]. Finally, items similar to user preferred items are aggregated and ranked to generate recommendations.

2.2 Aggregation functions

To aggregate the items in ICF algorithms, several functions have been proposed. In this work we used the Reciprocal Rank Fusion (RRF) [4] and Collaborative Filtering Preference Aggregation (CFPA) [2] to combine items recommended by two different ICF algorithms.

The RRF combines document rankings from multiple ranked lists. RRF sorts the documents according to a naive scoring formula. Given a set D of documents and a set of rankings R , we compute the RRF score as shown in Equation 1.

$$RRFscore(d \in D) = \sum_{r \in R} \frac{1}{60 + r(d)} \quad (1)$$

The CFPA weights document similarity with the user rating for the document. Given a set D of documents, a user u , and a set S of similarities, we compute CFPA score for a given user as shown in equation 2.

$$CFPAscore(u) = \sum_i R(u, d_i) \times \vec{d}_i \quad (2)$$

In the formula, $R(u, d_i)$ is the rating that user u gave to document d_i and \vec{d}_i is a column of the document similarity matrix.

To combine the output of the two ICF algorithms we used the Weighted Arithmetic Mean (WAM) [2], as shown in equation 3.

$$WAMscore(u) = \frac{\alpha ICFA + (1 - \alpha) ICFB}{2} \quad (3)$$

2.3 Hybrid recommendation systems

In addition to collaborative filtering techniques, recommendation system can also be content-, demographic-, and/or knowledge-based [3]. These techniques can be combined in a unique hybrid system. Hybrid systems combine two or more algorithms to improve recommendations, overcoming limitations of individual algorithms. According to the classification given in [3], in this work we will use a weighted system characterized by combining numerically the score of different recommendation components.

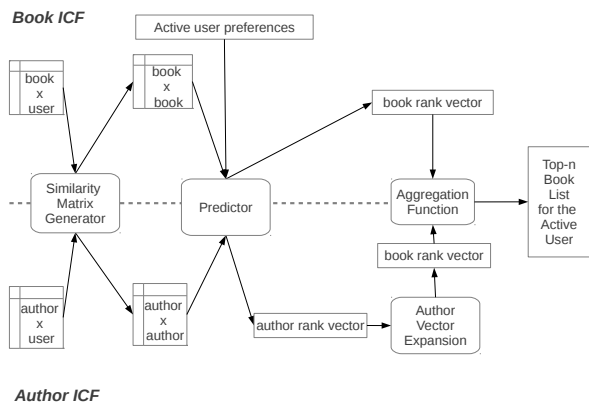


Figure 1: Hybrid book recommender algorithm.

2.4 Book and author recommenders

There are countless book recommendation sites that can be found on the Internet. Of these, we highlight: *gnooks*² (can recommend books and authors. *gnooks* includes a “literature map” that graphically shows authors read together); *Similar authors*³ (shows lists of authors similar to the author given user); *BookLamp*⁴ (defines the “book DNA” where author information is included and is used to find similar books). In [5], the author investigates the effectiveness of author rankings in a library catalog to improve book retrieval. However, to the best of our knowledge, this is the first study attempting to improve book recommenders through author recommendation.

3. MEAN RECIPROCAL RANK

To evaluate our approach we used the mean reciprocal rank (MRR). The MRR is a statistic used to evaluate the quality of top-n lists generated by retrieval processes. The MRR measures how far from the top appears the first good document and can be defined by Equation 4, where p_i is the position in the top of the first good document.

$$MRR = \frac{\sum_{i=1}^{\text{number of hits}} \frac{1}{p_i}}{\text{number of tops}} \quad (4)$$

where a *hit* is a well predicted document.

4. THE HYBRID BOOK RECOMMENDER

This section describes the LitRec data set and discuss our findings. Our hybrid book recommender (HBR) algorithm, outlined in Figure 1, is divided in three phases: the training or similarity matrix calculation, prediction, and aggregation phases.

4.1 LitRec data set

LitRec is a literary data set built for recommendation purposes. It combines documents from Project Gutenberg⁵

²<http://www.gnooks.com/>

³<http://www.similarauthors.com/>

⁴<http://booklamp.org/>

⁵<http://www.gutenberg.org/>

with ratings from *Goodreads*.

LitRec contains 38,591 ratings from 1,927 users and 3,710 documents. The data set also contains book authors (1,627 different authors), the user location (1,029 different locations), the review date, and document content. The review date was used to sort and divide ratings in a train-test set of 90%-10%.

4.2 Similarity matrix calculation

The HBR represents books as an N-dimensional vector of users where components contain the rating given by the user to the book. Ratings are integers in the 1 to 5 scale. Authors are represented as an N-dimensional vector of users where components contain the average of the ratings given by the user to the books written by the author.

The HBR generates book and author similarity matrices using book and author vectors and a similarity function. We made experiments using the cosine, the inverted Euclidean distance (Equation 5), and co-occurrences.

$$ieuc(\vec{u}, \vec{v}) = \frac{1}{\sqrt{(u_1 - v_1)^2 + \dots + (u_n - v_n)^2}} \quad (5)$$

Co-occurrences between book b_i and b_j are calculated by adding one to $cell_{i,j}$ of the $book \times book$ similarity matrix every time b_i and b_j are preferred by the same user. Co-occurrences between authors follow the same approach.

We also experimented using the cosine and Euclidean distance with 2nd-order vectors of co-occurrences [6]. 2nd-order vectors represent items as an N-dimensional vector of items where each component contains the number of times the items co-occur.

4.3 ICF prediction

Book and author rank vectors (RV) (Figure 1) are generated by the predictor using the similarity matrices and the active user⁶ (AU) preference vector. To generate the book RV, the predictor selects the $book \times book$ matrix columns corresponding to the user’s favorite books.

To calculate the author RV, the predictor counts the number of books that the AU preferred from each author. Then, it selects the $author \times author$ matrix columns corresponding to the user’s favorite authors.

The retrieved columns are aggregated using RRFscore (Equation 1) and CFPAscore (Equation 2). The main difference between the two scores is that CFPAscore weights the item columns with the user rating.

The evaluation results using the MRR statistic are shown in Figure 2. Evaluation shows that, for the LitRec data set, our algorithm produces better RV using co-occurrence matrices. However, author RV are more sensitive to user ratings than book RV.

As Figure 2 outlines, the best book predictions were achieved with co-occurrence matrix and RRFscore aggregation, whereas, the best author predictions were achieved with co-occurrence matrix and CFPAscore aggregation.

Compared to other data sets used for recommendation research, e.g., the Movielens⁷ data set, the number of ratings is much smaller. This generates very sparse rating matrices, leading to the poor performance of geometric similarity

⁶The active user is the user for which recommendations are being generated.

⁷<http://www.movielens.org>

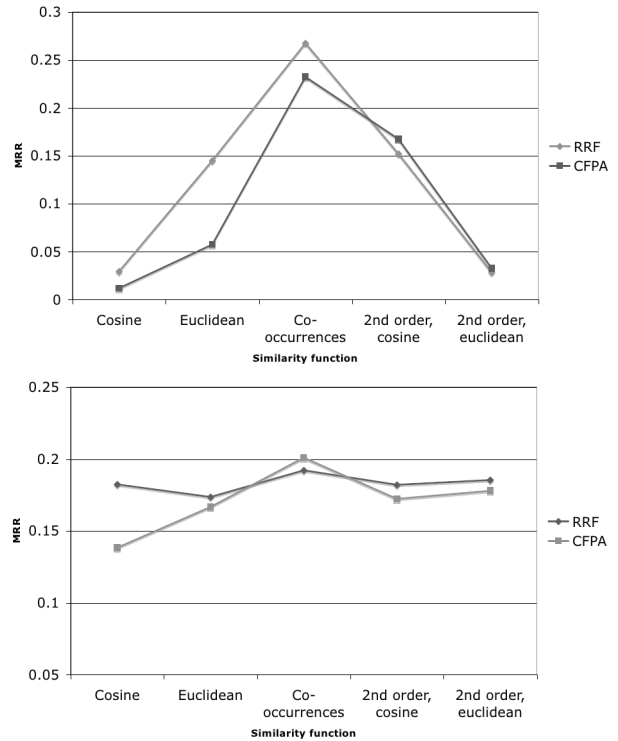


Figure 2: Book (top) and author (bottom) MRR comparing different similarity measures.

measures (cosine similarity and euclidean distance). In the remaining experiments, we used co-occurrence matrices to generate predictions.

4.4 Author vector expansion

After finding the authors similar to the AU favorite authors, the HBR expands the author RV to a book RV. The algorithm fills a book RV, assigning to each book its author rank weighted by the book popularity. Book popularity is measured by their frequency in the data set.

Experiments have shown that if the number of books per author is not restricted, the final book RV will be saturated by the authors with more books, leading to worse predictions. This led us to experiment with several book limits. The evolution of results is depicted in figure 3. As shown in the graphic, predictions improve when the book limit varies from 1 to 4 and decreases after 4 for both aggregation functions. The maximum number of books per author, for the LitRec data set, is 4. From here on we will a maximum of 4 books per author.

4.5 Aggregating book ranks

Both book RV obtained in the prediction step are finally aggregated. The Aggregation Function consolidates the book RVs into one single vector using the WAMscore (Equation 3). Then, we sort the final book vector, placing the most similar books at the top of the list. Finally, we select the top-n books with higher ranks, producing the top-n book list for the AU.

We varied the α parameter between 0 and 1 in order to assess the importance of the author in final recommendations and if final recommendations can be improved using

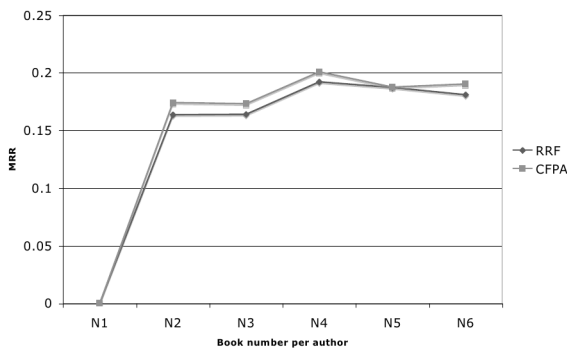


Figure 3: Limit on book number per author evolution.

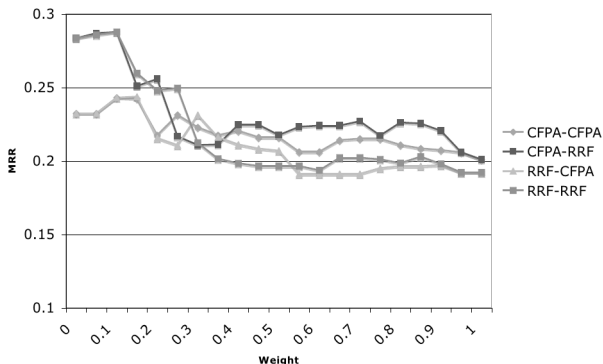


Figure 4: Hybrid book recommendation algorithm MRR for all combinations of author-book score aggregation. In the legend, the first aggregation score corresponds to the author and the second corresponds to the book.

the author. The evolution of results is shown in figure 4.

As shown in the graphic, the algorithm yields the best predictions when ranks have the combination of 10% author and 90% book. This means that the book information is much more important than the author information to predict the books that the user will like, but the author can contribute to improve the prediction.

The graphic also outlines the evolution of all combinations of score aggregations. As expected from results obtained in the previous experiments, when author predictions use CFPA score and books predictions use RRF score, overall results are better. However, at the 10%-90% author-book combination the RRF score-RRF score combination can achieve the same results.

5. CONCLUSIONS & FUTURE WORK

In this paper we describe a hybrid book recommendation algorithm. The HBR combines two ICF algorithms that predict the books and authors the user likes. Author predictions are expanded in to a book list that is subsequently aggregated with the former book list. Finally, the resulting book list is sorted to yield the top-n book recommendations.

The HBR was tested in the LitRec data set. LitRec data set has properties and limitations that can be found in a public library. This makes it suitable to study library sce-

narios and work out solutions that can be later adapted on a real public library.

The first of the initial goals was testing if ICF is suitable to predict books that the user will like in the LitRec conditions. Experiments led us to conclude that, the common ICF approaches yield poor predictions. When the algorithm uses co-occurrence matrices the first interesting books are placed near the third position in the book top-n. The second goal was to assess if book prediction by author selection can be used to improve overall predictions. Experiments in LitRec have shown that overall predictions can be improved using author prediction. However, a maximum number of books per author must be established, otherwise authors with more books will suffocate less productive authors, yielding one-author book top-n predictions. The maximum number for LitRec was set to 4.

We also observed that weighting the output of the both ICF algorithms differently achieve better predictions. For the LitRec data set, the contribution of choosing books by author must be smaller than book co-occurrences, i.e, book popularity. These will require experiments with a more extended combination of weights for further study.

This paper describes exploratory work in LitRec data set that open a path for further research. We intend to continue exploring LitRec. We will try to assess if book choice is related to content, user location, and the month in the book read date. Finally, the use of feature augmentation and dimensionality reduction techniques like singular value decomposition or principal component analysis will also be considered.

6. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734 – 749, June 2005.
- [2] G. Beliakov, T. Calvo, and S. James. Aggregation of preferences in recommender systems. In *Recommender Systems Handbook*, pages 705–734. Springer, 2011.
- [3] D. Burke. Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408, 2007.
- [4] G. Cormack, C. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759, New York, NY, USA, 2009.
- [5] J. Kamps. The impact of author ranking in a library catalogue. In *Proceedings of the 4th ACM workshop on Online books, complementary social media and crowdsourcing*, BooksOnline '11, pages 35–40, New York, NY, USA, 2011. ACM.
- [6] A. Kulkarni and T. Pedersen. SenseClusters: unsupervised clustering and labeling of similar contexts. In *ACL '05: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 105–108, Morristown, NJ, USA, 2005. ACL.
- [7] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web, WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.