As You Like It: Personalized Book Recommender System

Vansh Narula
Texas A&M University
College Station, Texas
narulavansh3@tamu.edu

Prakhar Mohan
Texas A&M University
College Station, Texas
prakharm007@tamu.edu

Manish Patel
Texas A&M University
College Station, Texas
manishp@tamu.edu

Ankita Singh
Texas A&M University
College Station, Texas
ankita_3096@tamu.edu

ABSTRACT

Most of the book recommender systems use Collaborative Filtering or Matrix Factorization to model the user's preference to the books. However one important feature that is typically ignored by recommender systems is the summary of the book itself. In this project, we propose a personalized book recommender system based on Bayesian Personalized Ranking (BPR) that uses summaries of books as one of the key features. We also, propose a novel way to modify user-user collaborative filtering using the summary embeddings as an additional weight of item-item similarity and comparing the results with the traditional collaborative filtering.

KEYWORDS

Book recommendation system, Collaborative filtering, Summary Embeddings, Bayesian Personalized Ranking

1 INTRODUCTION

A book recommender system predicts whether a user would be interested in reading a particular book or not. Avid readers can rely on such systems for books recommendations that cater to their personal interests. The most prominently used feature in these systems are the book title, author and genre of the book, for example fiction or fantasy. In this project, we explore the use of book summaries. This gives a better understanding of the whole content of the book and would thus be more informative in finding similar books according to user's taste.

Most Recommender Systems do not provide personalized suggestions as they learn from either explicit feedback such as star ratings or thumbs up/down data mostly using Collaborative Filtering [7]. Some recommender system do provide personalized suggestions using the implicit feedback such as browsing log, purchase history, hover activities etc. using Bayesian Personalized ranking or Matrix Factorization [6]. But When we are talking about a good book recommender, we need to learn a function that provides us the relation between the user and the book itself. Features such as book title, author name and genre of the book are too general to be treated as personal features to be used to learn that relationship. A user relates to the content of the book rather than its Genre, Title or Author name. None of the above mentioned techniques takes into account the content of the book and hence fails to incorporate the actual relation between the book and the user. Thus, failing on the personalized part that is much needed in a book recommender according to us.

In this project, we have developed model that incorporates summary embeddings which captures the contextual information of the book for modifying the current techniques to learn more personalized ranking function for each user-item pair. The main motivation behind this project comes from the VBPR paper of He et al. [5] where the authors had used the visual features of Amazon products for creating a better personalized ranking. Here we have used summaries of books to find similar books based on the content that have more chance of being liked by the user and have incorporated it into BPR method to create a new model which we have named SBPR(Summary-BPR) model following the trend.

We have also tried to incorporate these embeddings to modify the traditional Collaborative filtering model and see if the new model learns some personalized ranking based on the book content. For this we came up with a novel method that extends the traditional Collaborative filtering method to further refine the user's preferences for the items.

2 USER-ITEM COMBINED COLLABORATIVE FILTERING

The User - User collaborative filtering (CF) uses user-similarity as a weighting criteria for predicting ratings of a user u_i for an item i_a. While calculating this, the users that contribute in predicting this rating are the once that have actually rated the item i_a. This is represented by the equation given below.

$$r_{i,a} = \overline{r_i} + k \sum_{j=\#U} S(u_i, u_j).(r_{j,a} - \overline{r_j})$$

here $r_{i,a}$ is the rating of user i for item a that is to be predicted, $\overline{r_i}$ is the mean rating of user u_i, $\#U$ is the set of users nearest to the user i and S(u_i, u_j) denotes the similarity between the two users. When the data is huge, the interaction matrix formed is sparse and only a few users rate a particular item. Available users are further reduced as they are not among the most similar users to u_i. Therefore, this approach works good for user-item pairs for which enough data is available but it does not capture sufficient information for pairs with less data. Also, it fails to learn the relation between a user and an item. It is based on the premise that a user who is similar to the user in hand will be considered relevant only when it has rated the exact same item. This results in missing out on many potential similar users who might not have rated the exact same item. A user should be considered relevant not only when it has rated the exact same item but also when it has rated similar items to the item that we are considering.

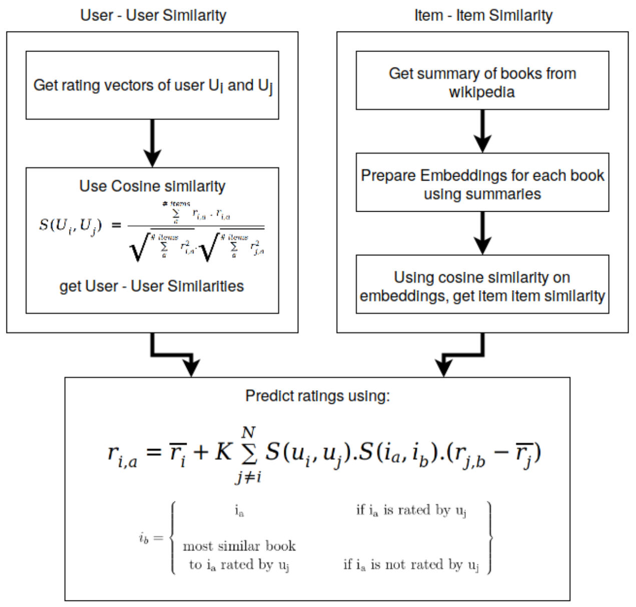
User - User Similarity
Get rating vectors of user Ui and Uj
Use Cosine similarity
$S(U_i, U_j) = \frac{\sum_a^{\# items} r_{i,a} \cdot r_{j,a}}{\sqrt{\sum_a^{\# items} r_{i,a}^2} \cdot \sqrt{\sum_a^{\# items} r_{j,a}^2}}$
get User - User Similarities
Item - Item Similarity
Get summary of books from wikipedia
Prepare Embeddings for each book using summaries
Using cosine similarity on embeddings, get item item similarity
Predict ratings using:
$r_{i,a} = \overline{r_i} + K \sum_{j \neq i}^{N} S(u_i, u_j).S(i_a, i_b).(r_{j,b} - \overline{r_j})$
$i_b = \begin{cases} i_a & \text{if } i_a \text{ is rated by } u_j \\ \text{most similar book to } i_a \text{ rated by } u_j & \text{if } i_a \text{ is not rated by } u_j \end{cases}$

Figure 1: Flowdiagram for rating prediction using User - Item combined Collaborative Filtering and book summaries as embeddings

To deal with this, we proposed a variation in user-user CF by introducing an additional weight of item-item similarity. The item-item similarity is calculated based on the content itself rather than the traditional item-item similarity based on user ratings. In our case it is taken as the similarity between the summary of the book in embedding space thus capturing the similarity of the content of the two books in lower dimension. With this, the algorithm is able to capture information even from those users who have not rated the item *a*. The algorithm in reference with book recommendations are shown in figure 1. Notice the item - item similarity term $S(i_a, i_b)$ introduced in the formula. Now, for every user *j* who has not rated the item *a*, the algorithm considers the item *b* which is most similar to the item *a* and is also rated by the user *j*. This way, if the user *j* has not rated the item *a*, his contribution is weighted down by the similarity of item *a* and *b*.

This can be understood with an example. Consider that the rating of user *i* is to be predicted for item "Harry Potter and the chamber of secrets". The user *j* has a similarity of 0.9 with user *i* but has rated "Harry potter and the philosopher's stone" which is 0.9 similar to the former. Simple User-User CF would have ignored this user for not rating the required book even though his rating can be used for prediction. Our proposed modification will use both the similarities and will weight the rating of user *j* for the later book by a factor of 0.9x0.9 = 0.81.

Thus, our proposed modification captures for information for predicting ratings by using user-user (ratings based) as well as item-item similarity (content based).

3 SUMMARY EMBEDDING BAYESIAN PERSONALIZED RANKING (SBPR)

Bayesian Personalized Ranking (BPR) is a pairwise ranking optimization framework that adopts stochastic gradient ascent as the training procedure [6] A training set D_s consists of triples of the form (u, i, j) where u denotes the user together with an item i about which they expressed positive feedback and a non-observed item j:

$$D_s = \{(u, i, j) | u \in U \wedge i \in I_u^+ \wedge j \in I \setminus I_u^+\} \tag{1}$$

The optimization criterion used for personalized ranking (BPR-OPT):

$$\sum_{(u,i,j) \in D_s} \log \sigma(\hat{x}_{uij}) - \lambda_\theta ||\theta||^2 \tag{2}$$

where σ is sigmoid function,λ_θ is a model specific regularization parameter, θ is parameter vector of model class, and $\hat{x}_{uij}(\theta)$ denotes arbitrary function of θ that parameterises the relationship between the components of the triple (u, i, j).

When using Matrix Factorization as the preference predictor, $\hat{x}_{uij}$ is defined as:

$$\hat{x}_{uij} = \hat{x}_{u,i} - \hat{x}_{u,j}$$

where $\hat{x}_{u,i}$ and $\hat{x}_{u,j}$ are defined by:

$$\hat{x}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i \tag{3}$$

Here α is global offset, β_u and β_i are user/item bias terms, and γ_u and γ_i are K-dimensional vectors describing latent factors of user u and item i. The inner product term $\gamma_u^T \gamma_i$ encodes the compatibility between the user u and item i i.e. the extent to which the user's latent preferences are aligned with the products' properties.

Our predictor model called **Summary BPR** takes the form

$$\hat{x}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T \theta_i \tag{4}$$

where α, β and γ are same as in Eq.3. θ_u, θ_i are newly introduced D-dimensional embeddings whose inner product models the **summary interactions** between u and i i.e extent to which the user u is attracted to each of the D dimensions. To transform high dimensional features i.e summary into lower dimensional space, we have used document embeddings generated using doc2vec api of gensim [3]. Figure 2 shows our implementation of Summary BPR method.

4 EXPERIMENT

In this project, we have used collaborative filtering and Bayesian Personalized Ranking to evaluate the importance of book summaries for book recommendation systems.

4.1 Dataset

The dataset used is 10K Goodbooks (Goodreads) which consists of 10,000 books with 6 million ratings by over 53,000 unique users. The dataset contains metadata of books like author, year of publication, genre, goodreads ID, etc. The explicit ratings include range from 1 to 5. The dataset was published on Kaggle in 2017 and can be accessed from [1]. For the purpose of this project, we have considered ratings of only 10,000 users. The training set has 976,982 samples and test set has 325,661 samples.

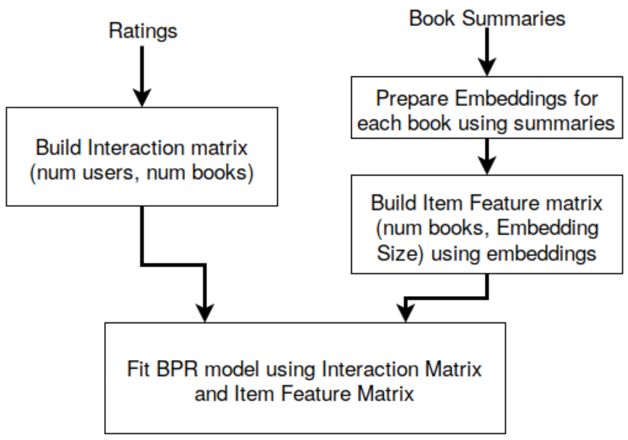
Ratings
Book Summaries
Build Interaction matrix (num users, num books)
Prepare Embeddings for each book using summaries
Build Item Feature matrix (num books, Embedding Size) using embeddings
Fit BPR model using Interaction Matrix and Item Feature Matrix

Figure 2: Flow diagram for Summary BPR.

4.2 Book Summary

The wikipedia-api [2] in python is used to find summaries/plots of the books of the dataset. This library allows searching on Wikipedia and parsing of data such as summaries, links, images, etc. A single string including the title of the book, along with the author and year of publication was used for the search. The command function 'wikipedia.summary()' searches for the Wikipedia page of that book and then returns the summary. If no Wikipedia page exists for a book then it is ignored in our project. This reduces the number of books from 10,000 to 6,176 books.

4.3 Summary Embeddings

Gensim [3] is a production-ready open-source library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. We have used Doc2Vec model from gensim to generate embeddings for books using the summaries captured through wikipedia-api. Each summary is represented by a vector embedding of length 125 and these vector representations are then used to find the similarities in the content of books.

4.4 Summary as a Feature

We have used the embeddings generated from summaries as feature in User-Item combined CF and S-BPR. For User-Item combined CF, the embeddings are used to find item item similarity. Item Item similarity is calculated as cosine similarity between two book's 125 dimensional embedding vector. These similarities are then used as explained under user-item combined collaborative filtering.

For implementing BPR, we have used lightFM [4] python library. LightFM is a Python implementation of a number of popular recommendation algorithms for both implicit and explicit feedback. For SBPR, the 125 dimentional embeddings are used as item features and a feature matrix of size (num_item x embedding_size) is made. This feature matrix is used by lightFM to train the BPR model.

5 RESULTS

Results in terms of MAE and RMSE for Collaborative Filtering are shown in table 1. We find that there is an improvement of 0.42% and 0.32% in terms of RMSE and MAE respectively, for the user-item collaborative filtering. This is attributed to the fact that for the users who have not rated the book instead of just omitting them, our model is trying to find the books which are similar to the book under consideration with the help of summary embeddings.
The results for Summary BPR model are shown in the table 2. We find an improvement of 0.76% and 1.1% in Precision@5 and Area Under ROC for the S-BPR over normal BPR. Some sample recommendations by our model are shown in the table 3.

Table 1: MAE and RMSE values for user-user CF and User-item combined CF

	User-User CF	User-Item CF
MAE	0.6983	0.6960
RMSE	0.8883	0.8845

Table 2: Precision@5 and AUC for BPR and S-BPR

	BPR	S-BPR
Precision@5	0.7575	0.7633
AUC	0.889	0.899

6 CONCLUSION AND FUTURE WORK

Summary of the book is an important feature since not only it paraphrase the main idea of the book but it also contains other features like genre, author etc. In this project we have explored the idea of converting summaries into embeddings and then using embeddings as a feature in Collaborative Filtering and BPR models. We propose a user-user Collaborative Filtering method that uses item-item similarity to find the users which may not have rated the same book but may rated the books that are similar to it. Similarly, we have implemented Bayesian Personalized Ranking using summary embeddings as a feature. The results show that our model outperforms the basic User-User CF model and Normal BPR methods.

However, we believe that we can still improve the performance of our methods. The hyper-parameter K can be replaced by weighted values of similarity scores but we did not have enough time to experiment with this idea. Moreover, for many books we could not generate summaries due to the lack of wikipedia pages or due to multiple pages-disambiguation error. We believe that usage of accurate summaries and new embeddings generation methods such as BERT can further improve the results. The source code of this project can be accessed from this [8] github link.

ACKNOWLEDGMENTS

This project was done as part of CSCE 670 coursework. We are thankful to Prof. James Caverlee , TA Jianling and our peers for their motivation and feed-back.

REFERENCES

[1] 10k-GoodBooks from GoodReads (https://github.com/zygmuntz/goodbooks-10k)
[2] Wikipedia-Api on python by Richard Asaurus (https://pypi.org/project/wikipedia/)
[3] Gensim (https://radimrehurek.com/gensim/)
[4] LightFM python library for BPR (https://lyst.github.io/lightfm/docs/home.html)

Table 3: Sample Recommendations

Users	Likes and Rates Books	Recommended Books
A	Night Trilogy	The House at Pooh Corner
	Lord of the Rings	Bad beginning
	100 years of solitude	Breathing Lessons
	Extremely Loud and Incredibly Close	The Stinky Cheeseman
B	Harry Potter and Goblet of Fire	Robinson Crusoe
	Harry Potter and Half Blood Prince	A Storm of Swords
	Survival in Auschwitz	The Boys in the Boat
	Story of Courage Community and War	The Stars are my destination

[5] He, R., & McAuley, J. (2016, February). VBPR: visual bayesian personalized ranking from implicit feedback. In Thirtieth AAAI Conference on Artificial Intelligence.
[6] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009, June). BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence (pp. 452-461). AUAI Press.
[7] Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative Filtering for Implicit Feedback Datasets. In ICDM (Vol. 8, pp. 263-272).
[8] https://github.com/manishpatel005/as-you-like-it