HYBRID BOOK RECOMMENDATION SYSTEM

TOLUNAY KATIRCI
tolunaykatirci@gmail.com

HAKKI ZAHİD KOCABEY
hakkizahidkocabey@gmail.com

FURKAN SARIYAR
sariyarfurkan@gmail.com

Supervisor: Prof. Dr. Çiğdem Eroğlu Erdem

Introduction

Problem

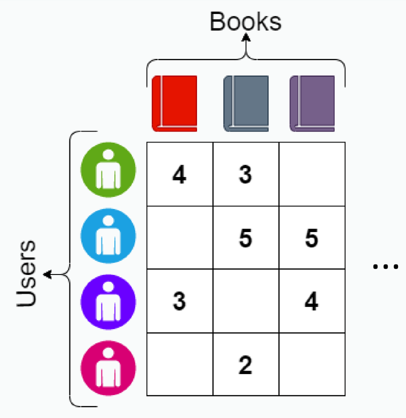
- With the increasing number of books, it becomes difficult to choose suitable books.
- Reading the back cover of the book may not be an effective way.

Solution

- Our goal is to build a recommendation system, which considers historical ratings of users and metadata of books.
- In order to give more accurate recommendations, we implemented a hybrid method in our recommendation system.

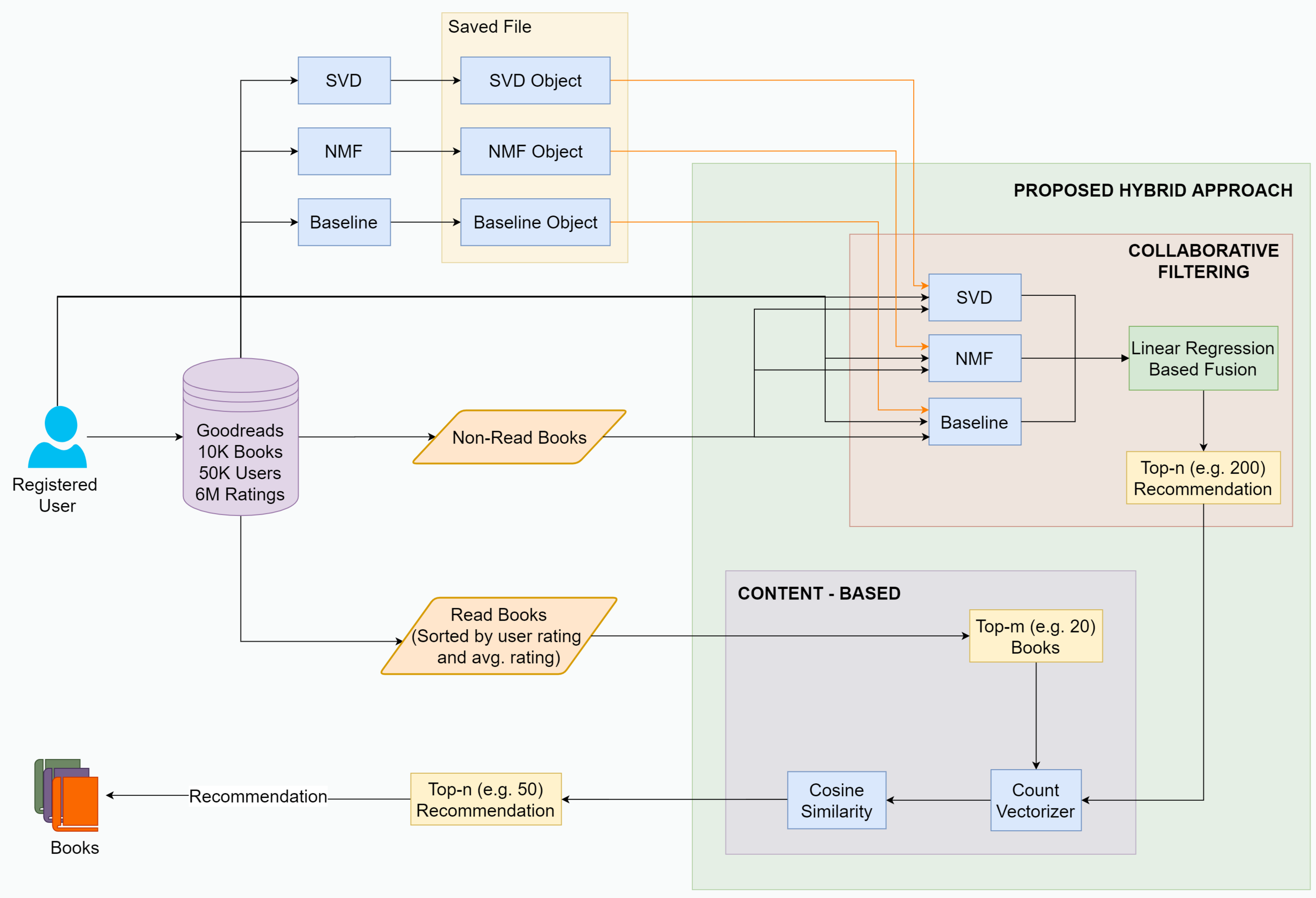
Proposed Hybrid Approach

- There are two main methods widely used in the recommender systems.
- **Collaborative Filtering** methods use a rating matrix.
- **Content Based** methods use metadata of books (e.g. author, genre).
- In our **Hybrid** approach, we first fused 3 collaborative filtering methods and then applied content based methods to improve the results.

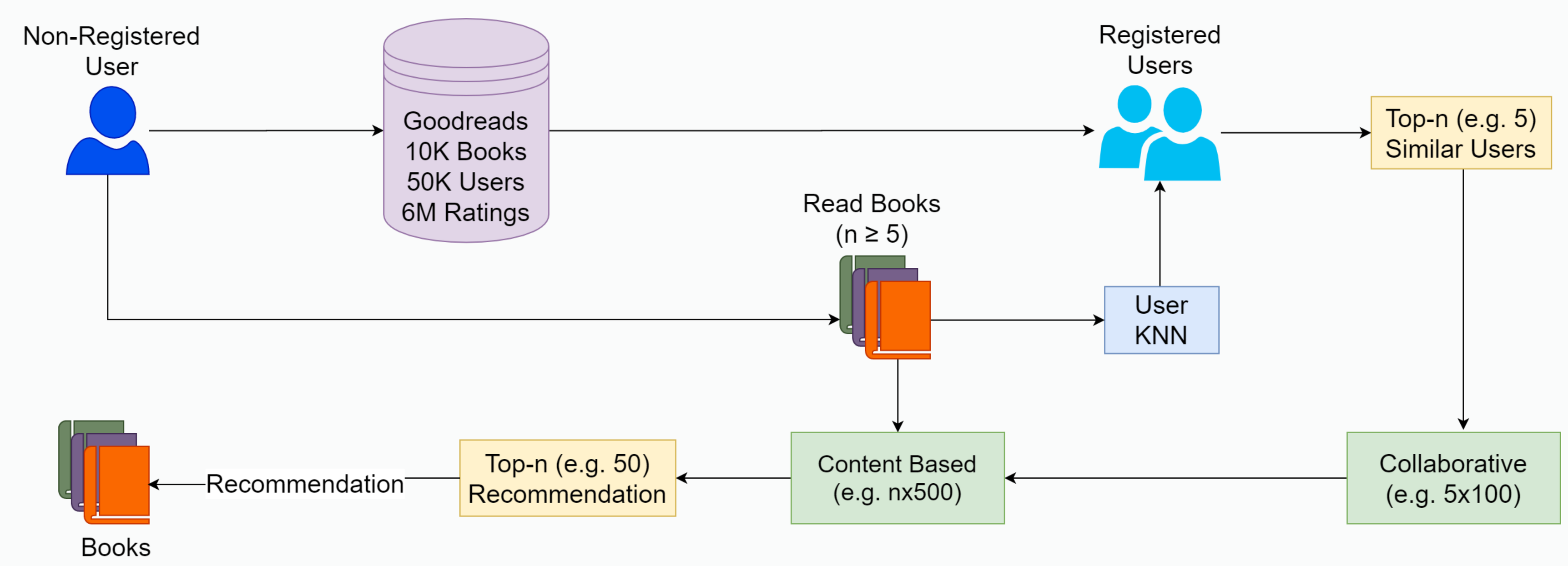


Rating Matrix (m*n)

Proposed Method for Registered User



Proposed Method for Non-Registered User



Algorithms

Baseline Algorithm

- All users and items have bias.
- Try to find the most appropriate value for biases.
- Update biases in each iteration.

$$\hat{r}_{ui} = \mu + b_u + b_i$$
$$e_{ui} = r_{ui} - \hat{r}_{ui}$$
$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$$
$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$$

$\hat{r}_{ui}$: Estimated element of rating matrix

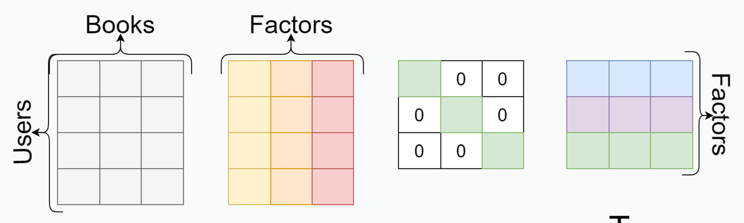
b_u : User bias

b_i : Item bias

Algorithms

Singular Value Decomposition (SVD)

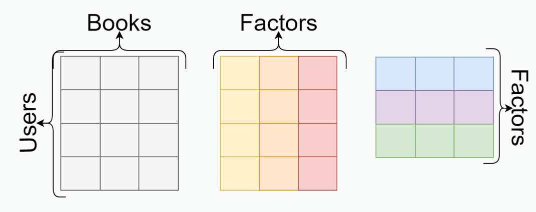
- Decompose Rating Matrix into three matrices.
- Try to fill empty cells by recomposing matrix M.



$$M_{mxn} = U_{mxk} \; S_{kxk} \; V^T_{kxn}$$

Non-Negative Matrix Factorization (NMF)

- Decompose Rating Matrix into two matrices.
- Try to fill empty cells by recomposing matrix M.
- All values in matrices are non-negative.



$$M_{mxn} = W_{mxk} \; H_{kxn}$$

Linear Regression Based Fusion

- Combine all Collaborative Filtering algorithms using linear regression to get a more accurate rate estimate.
- Regression coefficients are learned using validation set.

$$\hat{r} = \alpha_0 + \alpha_1 \hat{r}_{SVD} + \alpha_2 \hat{r}_{NMF} + \alpha_3 \hat{r}_{BL}$$

Count Vectorizer

- Basically counts given features (authors, genres) in specified documents.
- Select Collaborative Filtering results, which have most similar attributes (authors, genres) to read books to get more accurate results.

K-Nearest Neighbor

- Find most similar users to current user by applying cosine similarity using rows of rating matrix.

Experimental Results

Data Sets

Dataset Name	Number of Users	Number of Items	Number of Ratings	Density
GoodBooks-10K	53,424	10,000	5,976,479	0.011
MovieLens-1M	6,040	3,952	1,000,000	0.042

Rating Metrics

MovieLens 1M

Methods	RMSE
NMF	0.910
SVD	0.858
Base Line	0.907
I-Autorec [1]	0.831
Sparse FC [2]	**0.824**
RNMF [3]	0.871
MIXD [3]	0.861
Proposed Hybrid	0.855

GoodBooks-10K

Methods	RMSE
NMF	0.859
SVD	0.841
Base Line	0.856
Co-Clustering [4]	0.873
Slope One [4]	0.856
Proposed Hybrid	**0.839**

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(r_{ui} - \hat{r}_{ui})^2}{n}}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Classification Metrics

- Item with a rating value higher than 4 in the test set are considered as a relevant item.
- Precision calculates how many recommended items are relevant.
- Recall calculates how many related items are recommended.
- F1 score gives harmonic mean of precision and recall values.
- The system is better when these values are higher.

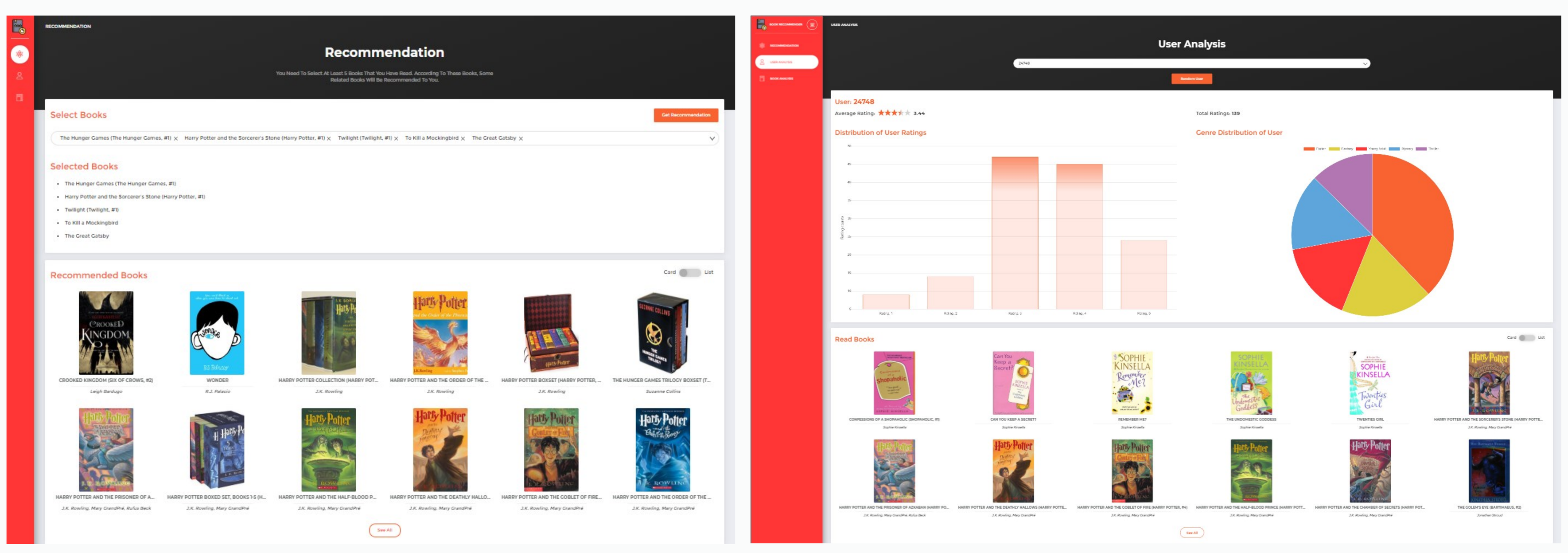
MovieLens-1M

Methods	Precision@10	Recall@10	F1@10
TC-CML [5]	0.66	0.13	0.22
Variant [6]	0.27	0.13	0.36
Proposed Hybrid	**0.67**	**0.76**	**0.66**

GoodBooks-10K

Methods	Precision@10	Recall@10	F1@10
Proposed Hybrid	**0.70**	**0.49**	**0.55**

User Interface of Book Recommendation System



Conclusion

- Designed a hybrid book recommendation system, which combines collaborative filtering and content based methods in a novel way.
- One of our contributions is Linear Regression Based Fusion of 3 collaborative filtering results.
- Our results are better than state-of-the-art in the terms of RMSE on Goodbooks-10K dataset.
- Recommendation system gives a dataset coverage of 23% for k=10.
- As a future work, coverage can be improved by recommending unrecommended books with high ratings.

References

[1] S. Sedhain, A. K. Menon, S. Sanner, L. Xie, Autorec: Autoencoders meet collaborative filtering, in: Proceedings of the 24th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2015.

[2] L. K. Muller, J. N. P. Martel, G Indiveri, Kernelized Synaptic Weight Matrices, in: Proceedings of the 35th International Conference on Machine Learning, 2018.

[3] G. M. Del Corso, F. Romani, Adaptive nonnegative matrix factorization and measure comparisons for recommender systems, in: Applied Mathematics and Computation 354, 2019

[4] "Algorithms Comparison" https://github.com/dorukkilitcioglu/books2rec [Accessed: 20/12/2019]

[5] B. Paudel, S. Luck, A. Bernstein, Loss Aversion in Recommender Systems: Utilizing Negative User Preference to Improve Recommendation Quality, in: Proceedings of The First International Workshop on Context-Aware Recommendation Systems with Big Data Analytics (CARSBDA), 2019

[6] J. Wilson, S. Chaudhury, B. Lall, P. Kapadia, Improving Collaborative Filtering based Recommenders using Topic Modelling, 2014

Technologies Used

surprise
learn
PostgreSQL