Book Recommendation System Using Distributed Item–Item Collaborative Filtering

Vadim Sokolov
Master's Degree in Computer Science

Algorithms for Massive Datasets – Project

1 Introduction

This project implements a book recommendation system using large-scale user review data. The system is designed following techniques studied in the course, in particular collaborative filtering implemented on distributed data processing frameworks.

The goal is to recommend books to users based on past ratings while handling datasets that cannot be processed efficiently on a single machine without distributed techniques.

The system is implemented in Google Colab using PySpark and processes Amazon Books Reviews data from Kaggle.

2 Dataset

We use the *Amazon Books Reviews* dataset available on Kaggle. It contains user reviews and ratings for books.

Each rating record contains:

- user identifier
- book identifier
- rating score (1–5)
- book title

After cleaning invalid ratings and removing missing values, we randomly sample 20% of the dataset for efficient experimentation.

After filtering users and books with very few ratings, the resulting dataset contains:

- 85,603 ratings
- 9,795 users
- 13,660 books

3 Method

We implement an **item–item collaborative filtering** recommender system.

The pipeline is:

1. Group ratings by user.

2. Generate pairs of items rated by the same user.
3. Compute cosine similarity between items.
4. Keep top-K neighbors per item.
5. Predict user ratings using weighted averages of similar items.

Similarity between items i and j is computed as cosine similarity:

$$sim(i,j) = \frac{\sum_u r_{ui} r_{uj}}{\sqrt{\sum_u r_{ui}^2}\sqrt{\sum_u r_{uj}^2}}$$

Predicted rating for user u on item i:

$$\hat{r}_{ui} = \mu_u + \frac{\sum_j sim(i,j)(r_{uj} - \mu_u)}{\sum_j |sim(i,j)|}$$

where μ_u is the user mean rating.
This mean-centering reduces user bias effects.

4 Implementation Details

Key parameters:

- Sampling fraction: 20%
- Minimum common users per item pair: 2
- Neighbors per item: $K = 30$
- Maximum items per user considered: 50

All processing steps are executed using Spark RDD transformations and aggregations.

5 Evaluation

We split the dataset randomly:

- 80% training
- 20% testing

Evaluation metrics:

- Root Mean Square Error (RMSE)
- Coverage (fraction of predictions produced)

Results:

Method	RMSE	Coverage
Global mean baseline	1.037	1.00
Item-item CF	0.422	0.375

The collaborative filtering method significantly improves prediction accuracy compared to the baseline.

6 Example Recommendations

For active users, the system recommends books consistent with their preferences, for example classic literature readers receive recommendations for similar classic works.

The system also provides explanations based on similar books previously rated by the user.

7 Discussion

The dataset is strongly skewed toward 5-star ratings, making rating prediction less informative in some cases. However, ranking-based recommendations remain meaningful.

Coverage is limited by sparsity, since many books share few common reviewers.

Future improvements could include:

- matrix factorization methods
- hybrid content-based approaches
- improved similarity regularization
- better cold-start handling

8 Conclusion

We successfully implemented a distributed recommendation system using item-based collaborative filtering. The system scales to large datasets using Spark and produces meaningful recommendations while improving over baseline prediction methods.