Project Report: Rating Prediction Using User-Based Collaborative Filtering (UBCF)

1. Introduction

Recommender systems are essential tools in modern digital platforms, enabling personalized suggestions for users based on their preferences and behavior. This project focuses on implementing a rating prediction algorithm using User-Based Collaborative Filtering (UBCF) methods in C++.

2. Problem Statement

Recommender systems predict user preferences for items such as movies. The goal of this project is to predict the rating a user might give to a particular movie based on historical ratings. Given a dataset containing user-movie ratings, the task is to estimate the missing ratings using UBCF. The project evaluates predictions using the Root Mean Squared Error (RMSE) metric, ensuring both accuracy and computational efficiency.

3. Algorithms and Methodology

3.1 User-Based Collaborative Filtering (UBCF)

UBCF predicts a user’s rating for a movie by analyzing ratings from similar users. The steps include:

1. Calculate the average rating of each user.
2. Identify similar users based on mean-centered cosine similarity.
3. Predict the rating as a weighted average of ratings provided by similar users.

3.2 Mean-Centered Cosine Similarity

This metric measures the similarity between two users by comparing their mean-centered ratings:

$$Similarity(A,B) = \frac{\sum_{i \in I} (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i \in I} (A_i - \bar{A})^2} \sqrt{\sum_{i \in I} (B_i - \bar{B})^2}}$$

Where:

- Ai and Bi are the ratings for item i by users A and B.
- A¯ and B are the average ratings of users A and B.
- I is the set of commonly rated items.

3.3 Prediction Formula

$$\hat{r}_{u,m} = \bar{r}_u + \frac{\sum_{v \in S} \text{sim}(u,v) \cdot (r_{v,m} - \bar{r}_v)}{\sum_{v \in S} |\text{sim}(u,v)|}$$

The predicted rating for a user u and movie m is:

Where:

- r^ u,m: Predicted rating.
- rˉu : User u’s average rating.
- S: Set of similar users.
- sim(u,v): Similarity between users u and v.
- r v,m: Rating given by user v to movie m.

4. Implementation

4.1 Data Structures

- **unordered_map<int, unordered_map<int, float>> rating_map**: Stores user ratings for movies.
- **unordered_map<int, float> user_average_ratings**: Stores average ratings of each user.
- **Vectors**: Used to store similarity scores and test data.

4.2 Key Functions

- **calculateUserAverageRatings**: Computes the average rating for each user.
- **meanCenteredCosineSimilarity**: Implements mean-centered cosine similarity to calculate user similarity.
- **predictRating**: Predicts the rating for a given user-movie pair by considering the top kk most similar users.

4.3 Workflow

1. Parse the dataset to separate training and test data.
2. Compute average ratings for all users.
3. For each test pair, compute similarity with other users and predict the rating.
4. Output the predictions.

5. Evaluation

5.1 Metrics

The performance is measured using Root Mean Squared Error (RMSE).

5.2 Results

Initial testing showed that the implementation achieved an RMSE below 1.0, indicating good prediction accuracy. Optimizations in data handling further improved performance for larger datasets.

6. Challenges and Improvements

6.1 Challenges

- **Computational Complexity**: Calculating similarities for large datasets was time-intensive.

6.2 Improvements

- Experimenting with alternative similarity metrics (e.g., Pearson correlation).
- Implementing User-Based Collaborative Filtering (UBCF) for comparison.
- Optimizing similarity computations using advanced data structures.

7. Conclusion

The project successfully implemented UBCF for rating prediction, achieving high accuracy with reasonable computational efficiency. Future work includes exploring hybrid approaches and further optimizations for scalability.

Team Members:

- Mehmet Efe Çakır
- Ömer Faruk Tokgöz
- Adil Bora Binici
- Halil Özer