Programming Assignment 1: Recommender Systems

Advances in Data Mining

Introduction

This assignment focuses on implementing different approaches for building recommender systems, following the material presented in Lecture 2. You will be working with user-movie ratings data, contained in the `u_data` file attached to the assignment, and your task will be to compute cosine similarities between users or items, implement collaborative filtering algorithms and work with matrix factorization methods.

Task Descriptions

1. Cosine Similarity

Implement a function to compute the cosine similarity between users (`axis = 0`) or items (`axis = 1`) in a user-item matrix.

- Function: `similarity_matrix`
- You must manually compute the cosine similarity according to this formula:

$$cosine\ similarity(A,B) = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2 \cdot \sum_{i=1}^{n} b_i^2}}$$

- Return a dictionary where each key is a user (`axis = 0`) or item (`axis = 1`), and the value is a list of the top-*k* most similar users or items based on the cosine similarity scores.
- No external libraries that directly calculate cosine similarity (such as `sklearn` or `numpy` built-ins) are allowed. You are allowed to use helper functions, for example, to compute dot products.

2. Collaborative Filtering

Using the cosine similarity matrix between users (or items), implement user-based collaborative filtering to predict the rating of a user for a given movie, and item-based collaborative filtering to predict the rating of a movie given a user.

- Functions: `user_based_cf`, `item_based_cf`
- Predict the rating for a user or a movie using this formula.

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_k(u)} (similarity(u,v) * r_{v,i})}{\sum_{v \in N_k(u)} similarity(u,v)}$$

3. Matrix Factorization

Complete the implementation of matrix factorization using the Gradient Descent Algorithm presented in the slides.

- Function: `matrix_factorization`
- Follow the provided implementation and try to understand how this algorithm actually works and how it can be used to address real tasks.

Submission

You have been provided with an `assignment.py` file that contains the skeleton code for the assignment. Go through the code and read the comments that should give you clear indications about the functions where you need to write your own code, specifying which arguments each function receives as input and which it should return as output.
Please don't delete or modify any part of the skeleton code, but just add your code in the sections where it is requested.
Submit the file `assignment.py` containing your implementation of the following functions:

- `similarity_matrix`
- `user_based_cf`
- `item_based_cf`
- `matrix_factorization`

Ensure that all your code is well-commented and follows proper Python standards. You are not allowed to import any external libraries other than those specified in the file. To ensure a correct submission, run the `validate_submission` test. This tests against all common submission errors and ensures that the submitted file can be evaluated. If the test fails, the evaluation will also fail, resulting in a **fail** for the assignment, so make sure the test passes when submitting your work.

Dataset

You will use the `u.data` dataset, which contains user ratings for movies on a streaming platform. You can find this file attached to the assignment, please don't modify this. In the code, to correctly import the dataset, please ensure that you replace the variable `path` with the correct path to your dataset.

Additional Notes

Make sure to test your functions with different test cases to ensure correctness. You can find a `main` section to test your implementations at the end of `assignment.py`.

Submission

The submission file should follow the naming convention `assignment_<xxxxxxx>.py`, where `<xxxxxxx>` is your student number without the leading ’s’. For example, for student number s1005819, the submission file should be named `assignment_1005819.py`. You should submit a single .py file. Until the deadline, you can update your submission. Only the last submission will be evaluated.

Good luck with the assignment!