INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Big Data Algorithms

TP6: Movie Recommendations using Collaborative Filtering

We will use collaborative filtering to **predict a user's rating for a movie**, using the ratings of users who have already rated that movie and have a similar taste with the target user.

We will use as input the MovieLens database of real movie ratings, collected over a period of time. In particular, we will work with the `ratings.csv` file containing 100,000 ratings for 9,000 movies by 700 users. Each line in the file is formatted as follows:

```
<userID, movieID, rating, timestamp>
```

where the rating is a grade from 0 to 5.

We would like to predict the rating of user "1" for movie "31", so ignore this line in your input dataset. We will use Java.

Exercise 1. Finding similar users

First, identify the other users who have rated the movie "31". Then, for these users (+ user "1") build the utility matrix with users on the rows and all their rated movies on the columns. For the movies missing reviews use zeroes.

Compute the cosine similarity of user "1" with all other users in the matrix and retain the top k (let's say k=5) most similar users.

Exercise 2. Rating predictions

You can now compute the predicted rating of movie "31" for user "1". Use a very simple predictor: the average rating of the top k most similar users.

Exercise 3. Fine tuning

Let's refine this prediction. Use a finer grained predictor: the average ratings of the other users, weighted with the similarity value.

Remember that, when using cosine similarity, a missing rating is considered like a negative rating. So, normalize the ratings of each user by subtracting their average rating from all its ratings (like this, all the ratings are now centered at 0). Recompute the prediction.

Increase the number of k most similar users to consider and see if the prediction improves.

Finally, replace the cosine similarity function with the Jaccard one and compute the prediction. What do you notice?

Exercise 4 (optional). Predictions at large scale with Flink

The MovieLens database contains larger scale datasets: the full database has 26,000,000 ratings for 45,000 movies by 270,000 users and is available here (224 MB):
https://grouplens.org/datasets/movielens/

Write a Flink program that implements the previous exercise using this dataset. You can exploit some built-in constructs to ease the development. For instance, it is easy to identify the users who have already rated a movie using the `groupBy` operator.