International Core Journal of Engineering Volume 7 Issue 4, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0005

Collaborative and Content-Based Filtering with Data from MovieLen

Kaiyuan Lin[1], Bohao Li[2]

[1]University of Washington, School of Art and Science, Seattle, WA, 98195-3765, USA;

[2]Northwestern Polytechnical University, School of Computer Science, Xi'An, Shanxi 710072, China.

Abstract

With the data from MovieLens, Collaborative Filtering recommendation model, which is based on ratings from users, and content-based recommendation model, which is based on the tags, such as genres, of movies people have watched, are implemented to recommend movies to users with their records.

Keywords

Recommender System; Collaborative Filtering; Content-based Recommendation; Movie Recommendation.

1. Introduction

1.1 Introduction:

During quarantine, people need to stay at home. They are not able to go out for entertainment. In this situation, watching movies is an excellent relaxing choice. There are about half of a million movies all around the world, and how to find the exact movies you want is a necessary step starting an enjoyable movie time. Our team is going to explore recommenders for movies. In the work, collaborative and content-based filtering are separately implemented to recommend users movies.

1.2 Dataset

"The dataset contains 20000263 ratings and 465564 tag applications across 27278 movies. These data were created by 138493 users between January 09, 1995 and March 31, 2015, divided into 6 files. This dataset was generated on October 17, 2016. Users were selected at random for inclusion. All selected users had rated at least 20 movies." [1]

2. Models

2.1 Collaborative Filtering:

The core algorithm we use is Alternating Least Square.

This model is building a matrix on user and movies. The value of matrix are the ratings from users. The main purpose for the model is to fill up the "blanks" in the matrix, where users haven't rated the corresponding movies. To predict the value, we are basically minimizing sum of the square of the difference between the actual value and the predicted value. We will predict from a randomly negative initial value, calculate the square of difference, and then get a new value from the derivative of the user's matrix. Repeating the process until the difference being smaller than our presetting value or the iterative times reaching our presetting times.[2]

With those predicted values, we will sort the predicted ratings for the user and output the first ten highest-rated movies the user hasn't watched.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0005

2.2 Contented base filtering:

With this model, we are giving the recommendation based on tags of movies.

The first part is training. We build the table for movies and tags. Several randomly generated hash functions are separated into different groups to produce the candidate pairs: [Local similarity hashing]

Fitting the table into hash functions, the algorithm generates min-signatures for movies in different bands.[3]

Comparing the signatures of different pairs in each band by Jaccard similarity of min-signatures, we filter out those pairs with similarity bigger than 0.05, returning those pairs as candidates.

Then, we will go throw all candidate pairs to calculate the similarity and output those pairs and their similarities which are bigger than 0.05.

With those pairs, we can do recommendations:

We select the movies the user watched, finding all pairs from model above and sort them by similarity in descending order. In this step, we filter out the movies that the user has watched or gave rating 3 or lower. Finally, we take the first 10 movies of the highest similarity.

2.3 Evaluate

In order to evaluate the precision of our recommendation, we hide half of records from each user, and give recommendations on the left movies each user rated. The numbers of movies that are both in the hidden part and the recommendation part over the number of hidden movies for each user are the precision we get.

3. Conclusion

3.1 Analyze

For Collaborative Filtering, the recommendation makes no sense. For nearly all the users, the precision of this recommendation is 0. We think the problem here is the number of movies we have. The dataset for movies is really big, over 27,000 movies, but the rating scale is from 0 to 5, and most of the ratings are in the scale 3 to 5, so there must be amount of movies squeeze at the top of the rank, and only a few of them would be recommended because they are at the top in alphabetic order. In this case, the recommendation has nothing to do with precision.

For Content-based recommendation, the precision is better. The average precision is 0.5 per-person. Since it takes longer time to run the algorithm, our group randomly select 100 users as the samples to test. The problem for this recommendation is the bad stability. The precisions of recommendations for 10 users in the sample are below 0.2 and for another 30 users are above 0.6. We guess the problem comes from the few records we have when training the model.

3.2 Improvement:

We are thinking about doing a Collaborative Filtering first to improve the “quality” (the ratings) of candidate movies, and then following by a Content-based recommendation. In this way, the content-based recommendation would spend less space, and the stability of precision would be better.

3.3 Further Study

For each user, the movies that the user has watched will not be recommended to him.

For each movie, it may have different tags, and we can associate similar tags. For example, horror movies can be associated with inference movies. By doing this, we can recommend associated tags to users.

Also, users might be tired of watching a small group of movies, so providing several movies with totally opposite tags the user always watched would be helpful.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202104_7(4).0005

References

[1] GroupLens. (2018, August 15). MovieLens 20M Dataset https://www.kaggle.com/grouplens/movielens-20m-dataset

[2] Xiaohei, & user of zhihu. (2016, March 09). How to explain the principle of ALS algorithm in spark mllib? https://www.zhihu.com/question/31509438

[3] tijun. (2017) Collaborative filtering of spark-MLlib https://www.cnblogs.com/tijun/p/7880170.html