DS-GA 1004 Big Data – Final Project

Group 99

Adeet Patel, Alexandre Vives, Ilias Arvanitakis

Introduction

The purpose of this report is to utilize various methods that can be used to create a recommender system and evaluate their accuracy. We are going to use the latest version of the MovieLens dataset which comes in two versions. A small version that contains 9,000 movies and 600 users and a larger one with 58,000 movies and 280,000 users. The small version will be used for testing and optimizing the algorithm and the large dataset should be able to run efficiently without delays. Two models will be created. Initially we will use the popularity base model. Then we are going to employ the ALS approach. Both methods are going to be evaluated by the precision and MAP methods.

Data Preprocessing

Before implementing any model, exploring the data and partitioning them into training, testing and validation sub samples is the first thing that should take place. There are many ways we can partition the data, and our approach takes into consideration the way the predictions are made. The purpose of our model is to individually select 100 movies to recommend to each user within a set of users by using part of its historic data to find other users with similar ratings (and thus similar taste in movies).

First, we will split the users into three parts. The first part will contain 60% of the users and the remaining two have 20% each. To make recommendations for the users in the validation and test sets, our model needs some ratings from those users to be able to find similar users on the training set, therefore we still need some of their ratings in the training set. We decided to take 30% of the ratings of each user on the validation and testing sets and move them into the training set. Eventually, the training set will include all the ratings for 60% of the users and part of the ratings for the test and validation users. More details are available in our GitHub repository at the train_test_validation_script.py file.

Model Evaluation

The evaluation methods that we are going to use are the precision at k and MAP metrics. Precision shows us the average success rate of our per user recommendations for all the users. From the 100 suggested movies for every user, out of the movies that this user has seen we calculate the percentage of the movies that are found in the recommendation set. Finally, we average this across our users. The main difference of the MAP is that it penalizes the success rate of the recommendation, based on the order of the recommendation. The first recommendations are considered more important than the last ones and thus we get a better sense of our success rate using the MAP method.

Popularity Baseline model

The Popularity Baseline Model as the name suggests, will be used as a baseline to compare the more sophisticated ALS model. Essentially, using the training set we can find the 100 most popular movies across all the users. Using the test set and the precision at k as well as the MAP criteria, will help us establish our baseline, based on which we can make comparisons later. For the small dataset the baseline model has a precision at k of 2.26% and a MAP of 0.127%. For the large dataset the precision at k stands 0.156% and the MAP at 0.008356%. We see that both measures are considerably higher for the small dataset. This is reasonable since we only have 9000 movies and thus there is a higher chance of making correct recommendations just because we have a small pool of available choices. The large dataset has 58000 movies and as a result there is a lower chance of success when suggesting the same movie to everyone. Therefore, there is the need to use a more sophisticated approach that considers the preferences of the user.

Alternating Least Squares

In the real world where there is an enormous number of available movies and many users with different preferences, making the same recommendation to everyone is not the most efficient way. The ALS method is considering the history of every user and makes personalized recommendations. After running the model several times and tuning the hyperparameters, we concluded that we are going to use a rank of 75 and a regularization parameter of 0.02. After training the model in the small dataset we got a precision at k of 10.26% and a MAP of 2.86%. This is considerably higher than the Popularity Baseline Model. Concluding, in this case we see that making personalized recommendations increases the success rate of our model.