Collaborative Filtering Using the Netflix Data

Manpreet Kaur

Final Project

Fall 2021

Course: DSCI-6007

Instructor: Dr. Vahid Behzadan

Introduction

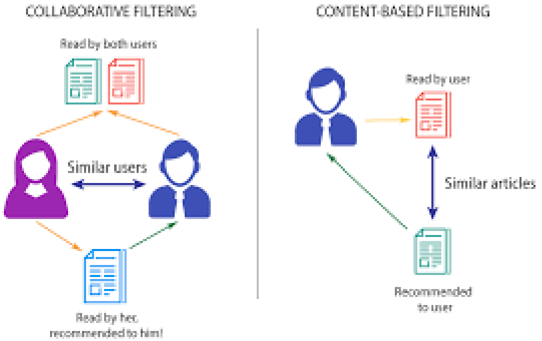
Recommender systems are the systems that are designed to recommend things to the user based on many different factors. These systems predict the most likely product that the users are most likely to purchase and are of interest to. Companies like Netflix, Amazon, etc. use recommender systems to help their users to identify the correct product or movies for them. The recommender system deals with a large volume of information present by filtering the most important information based on the data provided by a user and other factors that take care of the user's preference and interest. It finds out the match between user and item and imputes the similarities between users and items for recommendation. Both the users and the services provided have benefited from these kinds of systems. The quality and decision-making process has also improved through these kinds of systems.

Types of recommendation system

Content-based filtering methods are based on a description of the item and a profile of the user's preferences. In a content-based recommender system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended.

Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past.

Hybrid recommender systems which is combining collaborative filtering and content-based filtering could be more effective in some cases.



Alternating Least Squares

ALS takes a training dataset and several parameters that control the model creation process. To determine the best values for the parameters, we will use ALS to train several models, and then we will select the best model and use the parameters from that model in the project.

Project Set up

The goal of this project is to build a recommendation system for the NETFLIX data using SPARK in jupyter notebooks running on AWS EMR cluster with Data stored in a AWS S3 bucket .

1. Let's first see how to set the EMR Cluster in AWS account to analyze and implement the approach in a Jupyter notebook of EMR. Below are the steps to be followed:

- Login to the AWS Account and click on 'Amazon EMR'
- Create a EC2 key pair
- Click on 'Create cluster' and use the below configurations:
- Give a cluster name
- Choose Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.8.2
- Choose m4.xlarge or more
- Choose the EC2 key pair created previously
- Click on 'Create cluster'
- Wait till the cluster is created (till it shows 'waiting')
- Once the cluster is created, open the Security Groups for the Master node, and add 2
- inbound rules for SSH (anywhere) and Custom TCP Port 8888 (anywhere)
- SSH to the master node, the username is hadoop
- Run the below commands into the terminal:
- sudo pip3 install pyyaml ipython jupyter ipyparallel pandas boto -U
- export PYSPARK_DRIVER_PYTHON=/usr/local/bin/jupyter
- export PYSPARK_DRIVER_PYTHON_OPTS="notebook --no-browser --ip=0.0.0.0 --port=8888"
- Use the token and open in browser like this : http://ec2-XXX-XXXXXXXX.comput1.amazonaws.com:8888/tree?token=000111222333444555666777888999aaabbbcccdddeeefff#)
- The link opens the jupyter notebook
- Once the jupyter notebook is ready to use, the data can be imported using the s3 bucket created before with the datasets.
  Uploading the data into the S3 bucket:
- Login to the AWS Account and click on 'Amazon S3'
- Create a bucket and upload the dataset into the bucket
- This bucket can be accessed in the jupyter notebook created previously to load the data into the notebook and use it further for analysis

2. In this project, PySpark is used as coding language. Pyspark is the collaboration of Apache Spark and Python. Apache Spark is an open-source cluster-computing framework, built around speed, ease of use, and streaming analytics whereas Python is a general-purpose, high-level programming language.

The project follows the below approaches:

- User-based Collaborative filtering using ALS algorithm in the pyspark library
- Evaluation- Root Mean Squared Error

Evaluation Methods:

Root Mean Squared Error (RMSE)

MAE is average of the differences between values predicted by a model or an estimator and the values observed. Meaning, it is measure of difference between the Actual and predicted values. RMSE is just the square root of MSE. The predicted values can positive or negative as they under or overestimates the actual value. Squaring the residuals, averaging the squares, and taking the square root gives us the RMSE error.

The process we will use for determining the best model is as follows:

Pick a set of model parameters. The most important parameter to model is the rank, which is the number of columns in the Users matrix (green in the diagram above) or the number of rows in the Movies matrix (blue in the diagram above). In general, a lower rank will mean higher error on the training dataset, but a high rank may lead to overfitting. We will train models with ranks of 4, 8, and 12 using the training_df dataset.

Set the appropriate parameters on the ALS object:

The "User" column will be set to the values in our userId DataFrame column.

The "Item" column will be set to the values in our movieId DataFrame column.

The "Rating" column will be set to the values in our rating DataFrame column.

We'll using a regularization parameter of 0.1.

Have the ALS output transformation (i.e., the result of ALS.fit()) produce a new column called "prediction" that contains the predicted value.

Create multiple models using ALS.fit(), one for each of our rank values. We'll fit against the training data set (training_df).

For each model, we'll run a prediction against our validation data set (validation_df) and check the error.

We'll keep the model with the best error rate.

Implementation

1. Data Analysis and Approach planning

This dataset is a subset of the data provided as part of the Netflix Prize. TrainingRatings.txt and TestingRatings.txt are respectively the training set and test set. Each of them has lines having the format: MovieID,UserID,Rating. Each row represents a rating of a movie by some user. The dataset contains 1821 movies and 28978 users in all. Ratings are integers from 1 to 5. The training set has 3.25 million ratings, and the test set has 100,000. The files movie_titles.txt has rows having the format: MovieID,YearOfRelease,Title.

Some preliminary data analysis was done by looking at data summary statistics.

There are 3255351 train_ratings, 100477 test_ratings and 17769 movies in the datasets.

a) Distinct items and distinct users are there in the test set:
There are 27555 distinct users and 1701 distinct movies(items) in the test set.

b) (i) estimated average overlap of items for users
Three users were picked from the test set and extracted the items this user has rated in the training set. Then determined, how many other users in the training set have rated the same movies as the three target users chosen.
Target users from test set: 1447354, 534508, 992921.
The average overlap of items for first target user 534508 is: 8812
The average overlap of items for second target user 1447354 is: 12116
The average overlap of items for third target user 992921 is: 12477
The estimated average overlap of item for users is: 11135

(ii) **estimated average overlap of users for items**
Three target movies were picked from the test set and extracted the users that rated this movie in the training set. Then determined, how many other movies in the training set have been rated by the users who also rated target movie.
Target movies from test set: 481, 2366, 3149
The average overlap of users for first target movie 481 is: 303
The average overlap of users for second target movie 2366 is: 162
The average overlap of users for third target movie 3149 is: 215
The estimated average overlap of users for items is: 227

Collaboratives filtering approach lives from finding many similar users(user-user model) or many similar items(item-item model). User similarities are measured by overlapping items and item similarities are measured by overlapping users. From the above two statistics, the overlapping items for users is much higher(11135) than overlapping users for items(226).
Hence, I chose to implement the user-based collaborative filtering approach for the recommendation system.

2. **Building and Evaluating the Recommender**
   - (i) Split the Training data into training and validation set
   - (ii) Built the ALS model by setting the maxIter=5, regParam=0.1, userCol="UserId", itemCol="MovieId", ratingCol="Rating", coldStartStrategy="drop", implicitPrefs=False
   - (iii) Test the recommender on the test data set
   - (iv) Made top 5 recommendation for each user using our model

3. **Testing the Recommender on my preferences**
   - (i) Added some personal ratings for ten selected movies to the existing training data set to get a new combined training rating data set.
   - (ii) Trained a Model with the new combined ratings.
   - (iii) Evaluated the model for the new combined ratings.
   - (iv) Made top 5 movie recommendations for myself.

Conclusion

Collaborative filtering approach was used to predict the ratings of the Netflix users. User-user model for completed dataset was implemented using the ALS algorithm and the approach was evaluated using methods Root Mean Squared Error. Top five movies recommendations for each user was made using the model.