KEERTHI, PRANITHA

MOVIE RECOMMENDER SYSTEM

PROJECT REPORT

ABSTRACT

Recommender frameworks have turned out to be omnipresent in our lives. However, as of now, they are a long way from ideal. In this venture, I endeavor to comprehend the recommendation for a given movie id on the MovieLens dataset by using ratings given by the users. I endeavor to construct a versatile model to play out this examination. I begin by getting ready and contrasting the different models on a littler dataset of 1 million evaluations. At that point, I endeavor to scale the calculation, so it is ready to deal with 10 million 10 thousand evaluations by utilizing Apache Spark. For this examination I made use of Item-Based Collaborative Filtering algorithm.

1. INTRODUCTION

A recommendation system is a type of information filtering system which attempts to predict the preferences of a user and make suggests based on these preferences. Recommender frameworks are data separating instruments that try to foresee the rating for clients and things, dominatingly from enormous information to prescribe their preferences. Movie recommendation frameworks give an instrument to help clients in characterizing clients with comparable interests. This makes recommender frameworks basically a focal piece of sites and online business applications.

Recommender frameworks gather data about the client's inclinations of various things (e.g. motion pictures, shopping, the travel industry, TV, taxi) by two different ways, either certainly or unequivocally. A certain securing of client data regularly includes watching the client's conduct, for example, watched motion pictures, acquired items, downloaded applications. Then again, an immediate acquirement of data commonly includes gathering the client's past appraisals or history. Shared separating (CF) is the method for sifting or ascertaining things through the slants of other individuals. It first assembles the motion picture appraisals given by people and after that prescribes motion pictures to the objective client dependent on similarly invested individuals with comparable tastes and premiums before.

PROBLEM STATEMENT

Let *R* be the rating provided by the user with *userID* for a movie with *movieID*. For each userID map it's respective movieID and ratingID, i.e. for every user, each movie they rated are mapped together separately. Then every movie pair rated by the same user needs to be found and joined together which then helps to map for each movie, a pair of movies with their respective ratings and then it is used to compute similarity between ratings for each movie in the pair. This similarity is then used to display the recommended movies for a given movieID. The purpose is to find how well an existing algorithm in a recommender's system can find similarities in movies based on rating and genres. Can one large spark program be built to recommend similarity between movies?

GOAL

To find similar movies to each other based on the record based on movie ratings, i.e. finding similar movies using spark and the MovieLens dataset.

1.1 MOTIVATION

There is a wide assortment of utilizations for suggestion frameworks. These have turned out to be progressively mainstream throughout the last few a long time and are presently used in most online stages that we use. The substance of such stages shifts from films, music, books what's more, recordings, to companions and stories via web-based networking media stages, to items on web-based business sites, to individuals on expert and dating sites, to indexed lists returned on Google.

Frequently, these frameworks can gather data about a client's decisions, and can utilize this data to enhance their recommendations later on. For instance, Facebook can screen your cooperation with different stories on your feed so as to realize what kinds of stories claim to you. Now and then, the recommender frameworks can make enhancements dependent on the exercises of countless individuals. For instance, if Amazon sees that a substantial number of clients who purchase the most recent Apple Macbook likewise purchase a USB-C-toUSB Connector, they can prescribe the Adapter to another client who has quite recently added a Macbook to his truck.

Because of the advances in recommender frameworks, clients continually expect great suggestions. They have a low limit for administrations that are not ready to make proper recommendations. On the off chance that a music gushing application can't anticipate and play music that the client likes, at that point the client will just quit utilizing it. This has prompted a high accentuation by tech organizations on enhancing their proposal frameworks. Nonetheless, the issue is surprisingly perplexing.

2. RELATED WORK

Recommender systems are based on a variety of approaches such as content based, collaborative approach. Furthermost movie recommendation systems are centered on collaborative filtering and clustering. In movie recommender systems the user is asked to rate the movies which user has already seen then these ratings are applied to recommend other movies to the user that user has not perceived by utilizing collaborative filtering that is based on similar ratings. Collaborative filtering is tremendously spreading in such a way that this approach influences most of the recommender systems. Collaborative filtering majorly classified into two principal classes such as memory-based collaborative filtering and model based collaborative filtering. Memory-based collaborative filtering explores for nearest neighbors in the user space for an active user and dynamically recommend the movies. The shortcomings related to this method are computation complexity and data sparsity. Compared to this method research has shown that item-based method could decrease the time of computation as well as deliver rationally correct prediction and accurateness.

There are many web applications that are recommender systems which exist online. Few of them are,

- **FilmTrust**

FilmTrust is a site made as a piece of an exploration learn at the University of Maryland, which consolidates interpersonal organizations, motion picture evaluations and audits. Clients are urged to rate the amount they trust their companions motion picture taste on a scale from 1-10 (1 is low, 10 is high) and motion pictures on a size of 0-4, which is then used to compute anticipated evaluations for different motion pictures. in any case, the site covers the fundamental usefulness portrayed in the prerequisites list above and that's only the tip of the iceberg. A client can see a companion's motion picture appraisals and audits by visiting their profile or via hunting down a motion picture. Research anyway shows that clients will in general rate towards anticipated appraisals that are given by the framework (O'Donovan et al. 2008), which proposes that it would be better not to picture the anticipated appraisals. Golbeck (2006, p. 119), one of the authors of FilmTrust, states that clients can be delicate with regards to uncovering what trust esteems they relegate their companions, and on the grounds that the framework is dependent on the precision of this data it ought to be kept individual and undetectable from different clients.

- **Flixter**

Flixter.com is a social motion picture site and recommender framework. It is accessible on Facebook and is as of now a standout amongst the most prevalent applications with more than 17 million months to month clients (AppData, 2009). The application has a few choices for showing and rating motion pictures and additionally companion similarity tests, film tests, trailers and theater postings (for U.S. residents as it were). Flixter is likewise accessible on other long-range informal communication locales, for example, MySpace and Bebo and also a portable application for iPhone and Android. A client is given a few rating alternatives and additionally unequivocal data about how companions have evaluated films. Such unmistakable appraisals may have suggestions on how clients rate motion pictures. Ljung and Wahlforss (2008, p. 30) express that individuals when all is said in done tend to change how they present themselves relying upon what the circumstance is and who their gathering of people is. In the event that clients choose to rate motion pictures dependent on how they need others to see them the outcome may be that appraisals which don't generally mirror the client's taste are embedded into their evaluations framework, giving the framework a mutilated perspective of the client, perhaps bringing about awful proposals for the client. There is then again, a social part of having the capacity to see a companion's evaluations, much like having the capacity to see companions News Feeds, photographs and profiles (which are all Facebook highlights).

- **Jinni**

Another application that has been seen as motivation was the film recommender site Jinni.com. The site has been openly accessible in beta-adaptation since the seventh of October 2009 and keeping in mind that a considerable lot of their functionalities can't be utilized yet they have gotten a few grants for their innovation (Jinni 2009a). Motion pictures can be sought by substance where Jinni investigations film plots, disposition and then some, or they can be perused by inclination, plot, classification, time/period, put and different labels. Results are appeared as pictures which are estimated

by pertinence yet can be requested for example by span or rank. The outcomes can be additionally balanced on terms like "Little known" to "Well Known" or "Light" to "Serious" by hauling a slider. The site intends to sort films by distinctive qualities which has been finished by first letting film experts physically label motion pictures with whatever number viewpoints as could reasonably be expected and after that having their framework proceed with the procedure physically by gaining from the accessible labels (Jinni 2009c). It gives connects to TV-postings, DVD-rentals or internet spilling. Suggestions can be given to a client from the framework, comparative clients or express proposals from companions (Jinni 2009b). Proposals can likewise be given to a gathering of clients.

3. PROPOSED APPROACH

Two main approaches are widely used for recommender systems. One is content-based filtering, where user's interests' profile is created using information collected, and recommend items based on that profile. The other is collaborative filtering, where similar users are grouped together and the information about the group is used to make recommendations to the user. The approach used for this project is Collaborative filtering developed as a Spark problem.

ALGORITHM

This project was developed using Item-Based Collaborative Filtering.

ITEM-BASED COLLABORATIVE FILTERING

Item based collaborative filtering is a model-based algorithm for recommender engines. In the algorithm, the similarities between different items in the dataset are calculated by using one of a number of similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset. In item based collaborative filtering similarities between items are calculated from rating-matrix. And based upon these similarities, user's preference for an item not rated by him is calculated.

The process of item-based collaborative filtering can be explained in the following way,

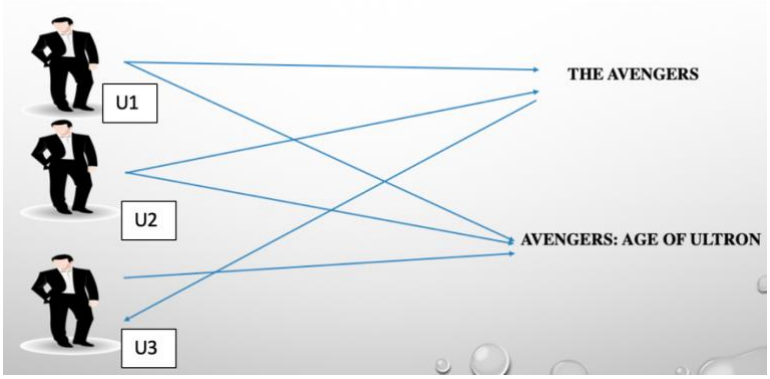
- Firstly, ever pair of movies that were watched by the same person are mapped together.
- Then, the similarity of their rating across all users who watched both is measured.
- Then sorting is done based on movies first then by similarity strength and the results is given.

One critical step in the item-based collaborative filtering algorithm is to compute the similarity between items and then to select the most similar items. The basic idea in similarity computation between two item i and j is to first isolate the users who have rated both of these items and then to apply a similarity computation technique to determine the similarity $s_{i,j.}$ There are a number of different ways to compute the similarity between items, cosine-based similarity, correlation-based similarity and adjusted-cosine similarity. This project uses cosine-based similarity to compute the similarity between movies.

Cosine-based Similarity:

In this, two items are thought of as two vectors in the *m* dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally, in the *m x n* ratings matrix, similarity between items *i* and *j*, denoted by *sim(i, j)* is given by

$$sim\ (i, j) = cos\ (i', j') = (i'.\ j') / (||i'||_2 * ||j'||_2)$$



For example, consider the above example where User1 and User2 have watched both The Avengers and Avengers: Age of Ultron and have also rated it. But, the User3 has only watched Avengers: Age of Ultron. When the item-based collaborative algorithm is applied to the above example it maps all the users with their movies and ratings respectively. After mapping for each user, the algorithm then joins the ratings and movies and finds the similarities between them. The result recommends User3 to watch The Avengers.

DATASET

This project was developed using MovieLens dataset to evaluate item-based collaborative filtering algorithm for recommending movies for the give movieID.

MovieLens

From the web-based site *grouplens.org* I made use of MovieLens 1M dataset to run my code locally and MovieLens 10M100K dataset to run my code on cluster (AWS). This dataset is generated from a web-based recommender system MovieLens.com where the users rate the movies anonymously.

MovieLens 1M dataset files contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000.

MovieLens 10M100K dataset file contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users of the online movie recommender service MovieLens.

All ratings are contained in the file "ratings.dat" and are in the following format:

UserID::MovieID::Rating::Timestamp

- Ratings are made on a 5-star scale (whole-star ratings only)
- Timestamp is represented in seconds since the epoch as returned by time (2)
- All users selected had rated at least 20 movies.

Movie information is in the file "movies.dat" and is in the following format:

MovieID::Title::Genres

EVALUATION

The step by step process to evaluate movie recommender system using Spark can be summarized in the following way,

- Firstly, the input ratings are mapped to (userID, (movieID, rating))
- Then every movie pair rated by the same user is found by a self-join operator, which results in (userID, ((movieID1, rating1), (movieID2, rating2))
- Then the duplicate pairs from the above result are removed and then the movie pairs are made the key, i.e. ((movieID1, movieID2), (rating1, rating2))
- Then grouByKey function is applied to above result to get every rating pair found for each movie pair.
- Then the similarity between ratings is computed for each movie in the pair which is then sorted and displayed as the top recommended movies for a given movieID.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

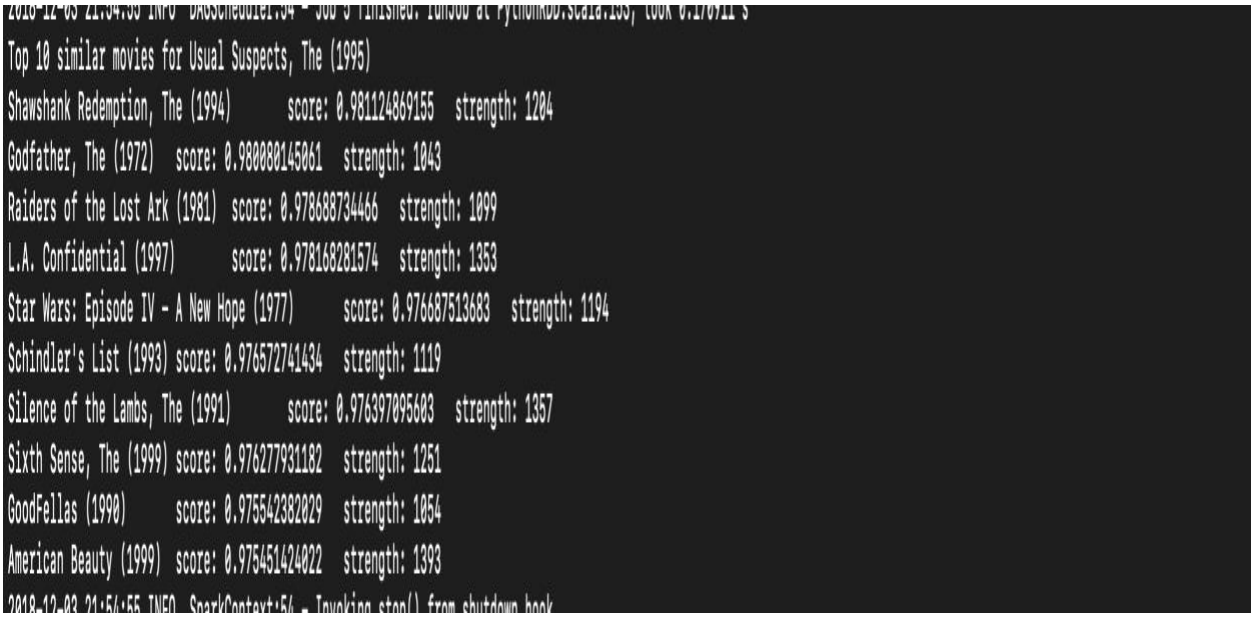


Figure 1: 1M Dataset on local machine

Figure 1 shows the result of top 10 recommended movies for the movie "The Usual Suspects" with movieID "50". This uses 1M MovieLens dataset from grouplens.org which is executed locally. The result displays the top 10 movies that are recommended for the users who have watched and rated "The Usual Suspects" with the score and strength of each movies similarity with "The Usual Suspects" movie.

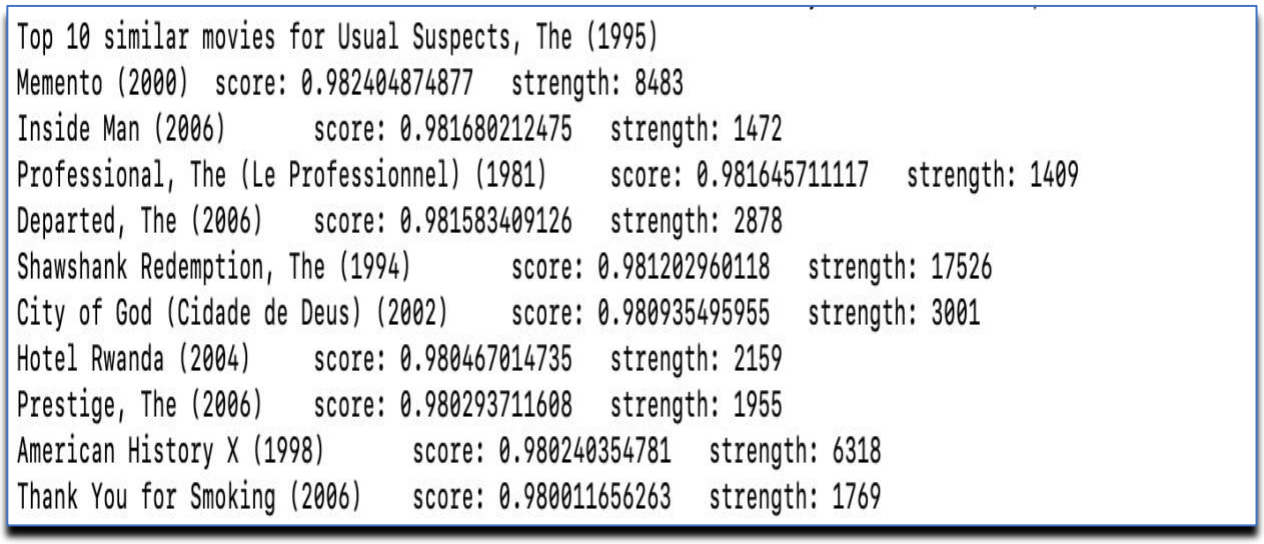


Figure 2: 10M100K Dataset on AWS Cluster.

Figure 2 shows the result of top 10 recommended movies for the movie "The Usual Suspects" with movieID "50". This uses 10M100K MovieLens dataset from grouplens.org. This dataset is huge and hence it can't be executed locally so I executed this dataset on the AWS cluster. The result displays the top 10 movies that are recommended for the users who have watched and rated "The Usual Suspects" with the score and strength of each movies similarity with "The Usual Suspects" movie.

The code was developed in a way to just display top 10 recommended movies, but the number of recommended movies can be altered too. Based on the performance, the 1M dataset took approximately 20 minutes to display the result when executed on local machine whereas the 10M100K dataset took approximately 50 minutes to display the result when executed on cluster.

5. CONCLUSION AND FUTURE WORK

In this article item-based collaborative filtering is applied to the MovieLens dataset to achieve a recommended system. The project outcomes Movielens dataset discussed indicated that for a given movieID the similar movies based on their similarity score is displayed. In real time the recommendations that were displayed were said to be true when asked to the people who watched Usual Suspects were the recommended movies satisfying.

There are plenty of way to expand to the work done on this project. Firstly, the project can be executed by discarding the bad ratings, i.e. only recommend good movies. Also, different similarity metrics like Pearson

Correlation, Jaccard Coefficient, and Conditional Probability can be tried on the approach. The thresholds for minimum co-ratings or minimum score can be adjusted to get more accurate results. A new metric that takes the number of co-raters into account can be invented and tested on the approach. And also, the genre items in "u.items" can be used to boost scores from movies in the same genre.

6. REFERENCES

[1] Collaborative Filtering Recommender Systems, *Stanford Student project*, http://cs229.stanford.edu/proj2014/Rahul%20Makhijani,%20Saleh%20Samaneh,%20Megh%20Mehta,%20Collaborative%20Filtering%20Recommender%20Systems.pdf

[2] Item-based collaborative filtering algorithms, *Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl*, https://dl.acm.org/citation.cfm?id=372071

[3] Research-Paper Recommender Systems: A Literature Survey, *Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger*, http://docear.org/papers/Research%20Paper%20Recommender%20Systems%20--%20A%20Literature%20Survey%20(preprint).pdf

[4] Collaborative filtering recommendation algorithm based on Haddop and Spark, *Bartosz Kupisz, Olgierd Unold*, https://www.researchgate.net/publication/282275966_Collaborative_filtering_recommendation_algorithm_based_on_Hadoop_and_Spark