Recommendation System for Sets of Movies

Simone Brentan
University of Trento
Trento, Italy

ABSTRACT

A recommendation system is a tool that suggests items(such as movies, music or products) to users based on their preferences and past behaviour. The goal of a recommendation system is to provide personalized recommendations to each user, in order to improve their experience and increase their engagement with the platform. For example, they can be used by music streaming services to generate playlists fitted for different users, or by search engines to predict possible future queries of a user.
Recommendation systems are also implemented by video streaming platforms and movie streaming services, such as the renowned Youtube and Netflix. In the last decade, the number of users using these types of applications has increased drastically, leading to an increase in the size of data that has to be processed.
This paper aims to provide a solution that uses a hybrid filtering technique to recommend sets of movies to individual users based on their preferences. An implementation of the solution is then provided and compared to the other most common approaches in this field, keeping in consideration the time and space consumption as well as the quality of the results.

KEYWORDS

Data mining, Hybrid recommendation system, Cosine similarity, IMDB movies dataset, Python, SVD, Hierarchical Clustering

1 INTRODUCTION

Recent studies found out that people are watching movies more frequently through streaming subscriptions than Pay-TV.[1] Even before the pandemic, another study discovered that young adults watch 2.5 times more internet video, such as the ones available on YouTube and Netflix, than TV.[4].
As the time goes by, and with the change of generations, the television will probably be taken over by online streaming services. The usage of platforms providing these type of services will increase exponentially and the data involved in their usual processes will impact the time and space required for their execution. The demand for efficient algorithms able to operate with such data is therefore ever more present.
Recommendation systems are services that aim to recommend relevant items, which vary depending on the field of application, to different users based on their preferences. Online streaming services need this systems in order to provide a better customer experience to users registered on their platforms. This paper aims to provide an algorithm for such a system that is able to recommend sets of movies, each of which is identifiable by a query on the database, by predicting how much the user desire to see its content.
The proposed algorithm is a hybrid recommendation system, which mixes aspects of both collaborative filtering and content-based filtering techniques. In order to make the prediction of query scores more efficient, the algorithm initially performs some precomputation on the data:

- Scores standardization
- Similar user candidates computation
- Similar query candidates computation

Scores standardization is needed in order to secure the homogeneity of the results, and helps to raise the score accuracy. As for the candidates computations, they aim to compute similar items so that only items with elements in common are checked instead of the whole dataset.
Both of them uses an algorithm of cosine similarity whose purpose is to compute only the top k similar indexes for every element through the usage of a sparse matrix. This algorithm[2] is said to be 40% faster and with reduced memory consumption compered to other classical methods.
In order to further reduce the computation time and storage usage, other algorithms were applied to the data:

- Dimensionality Reduction(SVD)
- Hierarchical Clustering

After having computed the candidates for both users and queries, the main prediction of scores is carried out. A collaborative filtering algorithm retrieves the queries most voted by similar users, while a content-based filtering algorithm retrieves a list of similar queries. Afterwords, a weighted average is applied to the queries and the result is returned. In addition, in order to resolve the problem of cold start, the total average scores are calculated and used instead of missing values.
To test the algorithm effectively, it was decided to use data most similar to real-world cases. For the list of movies, the IMDB movies dataset[3], containing nearly 45k movies, has been used. As for the list of queries and users, they have been generated randomly in an interesting way, so that they may represent useful information and resemble real data. The users' feedbacks were in fact decided based on some 'tastes' generated for every user, which could vary for example from genre, product company or popularity. In this way, all values are significant and, additionally, the results of the prediction algorithm can be tested accordingly. As for the content of the queries, they were mainly generated trying to avoid empty queries and to reduce the duplicated ones.

2 PROBLEM STATEMENT

This paper proposes a solution able to predict the values contained inside an utility matrix, which represents the scores the users have given to any query. Formally, the problem can be described as an algorithm which takes as input four files in CSV format containing respectively:

- the list of users
- the list of queries

- the utility matrix
- the list of movies

After its execution, the algorithm returns the filled utility matrix in the form of a file. In this way, it is simple for another application to fetch the recommendations to present to any user.
Every query is represented by a set of conditions, which describe the attributes that the movies need in order to be part of the answer set of that query. In this way the queries can be identifiable as a set of items, which in our case are movies. In the file representing the utility matrix, every user is associated with every query. The cells can contain a value between 1 and 100, which represents the score given to a query, or they can be empty, meaning the user has not given his feedback for that query.
The file containing the list of users and list of queries are used, additionally, to determine which user and query should be analyzed during the algorithm. If the utility matrix is of size M x N and U, Q are respectively the size of the users and queries files:

- 0 <= U <= M
- 0 <= N <= Q

This means that the users and queries files may be empty, in which case the algorithm will return an empty file, or can contain a maximum number of elements not bigger than the size dictated by the utility matrix. Furthermore, every id that is present in those files must as well as be present in the utility matrix.

REFERENCES

[1] 2022. People Are Watching Movies More Frequently Through Streaming Subscriptions Than Pay-TV. https://www.marketingcharts.com/industries/media-and-entertainment-225265
[2] 2022. Usage Statistics of Content Languages for Websites. http://w3techs.com/technologies/overview/content_language/all
[3] Rounak Banik. 2017. Hierarchical Matching Network for Heterogeneous Entity Resolution.
[4] Todd Spangler. 2016. Younger Viewers Watch 2.5 Times More Internet Video Than TV (Study).