Neural Network and Deep Learning Analysis Applied in Movie Recommendation System

ANLY 590 Group 3: Qian Leng, Xiaoning Ni, Yihui Liu, Yutong Du, Hanjun Wang

This project created movie Recommendation systems through different neural networks and deep learning models, including content-based model, collaborative filtering model, and behavior sequence transformer model. The datasets are gathered from [MovieLens] with over twenty five million user ratings from 162-thousand users over sixty thousand movies between January 09, 1995 and November 21, 2019. All selected users had rated at least 20 movies. Based on the historical rating of an individual user, the preference of the user can be predicted using a neural network algorithm. This project executed data gathering, data cleaning, data exploration, data training, model fitting, optimization and accuracy checking in the whole project.

Exploratory Data Analytics

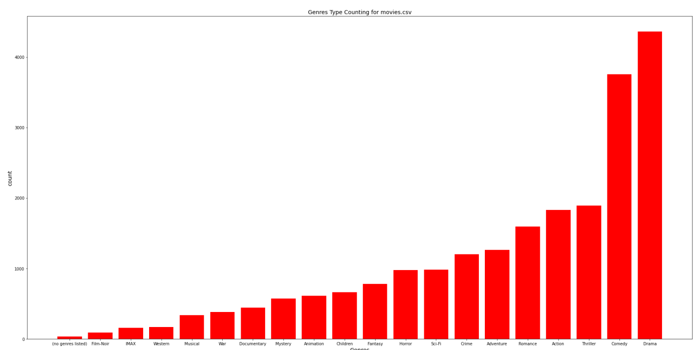

Figure 1. Genres type counting for movies.csv

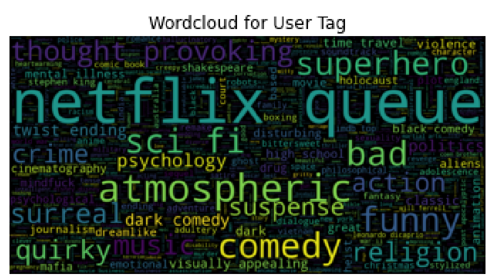

Figure 2. WordCloud for Users' Tag

Figure 1 is a bar chart for count of genres type in movies.csv. In this plot, it can be found that the most common movie genres are "Drama" and "Comedy", then "Thriller" and "Action". Which means, movie creators are more likely to make such genres and also may reflect the preference of customer markets. Figure 2 is a word cloud plot for users' tags(a short comment given after watching that movie). It suggests that Netflix does generate a few high quality movies and also superhero, scientific, atmospheric types and comedy are popular among the audiences.

Content-based model

The concept of the content-based model is to find the types of movies a certain user likes based on his/her historical ratings of other movies. After converting the original data into CSV format in the data procession notebook, a content-based model was applied to our MovieLens dataset to predict ratings. This particular system includes using tags, genres of the movies, and the user's historical rating to predict the rating and like/dislike of a given movie by this user (Figure 3). The system uses Random Forest to predict ratings using tags as the feature, while the Artificial Neural Network is applied to the model of the genres.The final ratings and like/dislike conclusion are given by a linear regression.

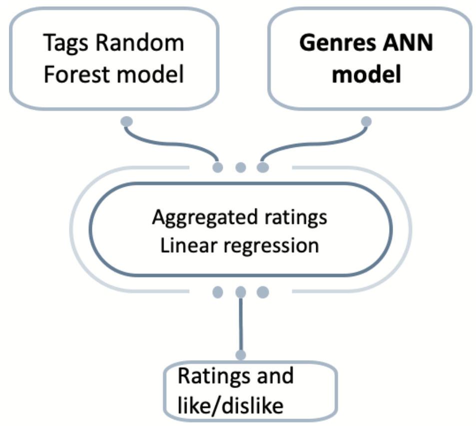

Figure 3. Structure of the content-based model

The genre model is feed forward neural network. First, it used two dense layers for the user-liked genres, two dense layers for user-disliked genres, and two dense layers for movies' genres. Then the model used a concatenated layer to combine the three layers. The output layer is a sigmoid forcing the output rating as a standard one, between 0 and 1. The results show that the training loss (MSE) is 0.039, while the validation loss is 0.040 for ratings. For the like/dislike category, the accuracy is 0.47.

There are pros and cons for the content-based model, as well as this particular model. For the pros, the model does not need data of other users as the only information it needs is the historical ratings for a particular user. Thus, the model can recommend unique movies for a particular user. And it is also easy to interpret and give the reasons of why the movies are recommended to a user, e.g. movie genre.

While the content-based model does not use other users' data, it is unable to help judgments of other users. Moreover, it doesn't support if there is a new user without any historical ratings. For movies with features the user has never watched before, the model would not recommend it to the user. For this particular model, the results highly depend on the labeling process, e.g. like/dislike, which requires considerable human judgment.

Collaborative Filtering Model

As discussed above, content-based recommender relies on the similarity of the items being recommended. The basic idea is that if you like an item, then you will also like a "similar" item. The system behind the collaborative filtering recommender is a little bit more complicated, it is entirely based on the past behavior and not on the context. More specifically, it is based on the similarity in preferences, tastes and choices of two users. It analyzes how similar the tastes of one user is to another and makes recommendations on the basis of that.

For instance, if user A likes movies 1,2,3 and user B likes movies 2,3,4, then they have similar interests and A should like movie 4 and B should like movie 1. This makes it one of the most commonly used algorithms as it is not dependent on any additional information.

For the CF model utilized for this project, firstly, some data preprocessing should be performed here to encode users and movies as integer indices. Secondly, both users and movies were embedded into 100-dimensional vectors. Thirdly, the model computes a match score between user and movie embeddings via a dot product, and adds a per-movie and per-user bias. The match score is scaled to the [0,1] interval via a sigmoid since the ratings are normalized to this range before. MSE loss function is generated to perform the result: for the training dataset, the loss is about 0.0437; while for the validation dataset, it is about 0.0439. However, only 8 epochs were used here because of limited time and computer units, it is likely that if epoch size was added, the model could perform better.

Figure 4. Training & Validation MSE by epoch

According to the recommendation outputs, the ratings dataset lists the ratings given by a set of users to a set of movies. The goal for the project is to be able to predict ratings for movies a user has not yet watched. The movies with the highest predicted ratings can then be recommended to the user.

Behavior Sequence Transformer Model

The BST model is a Transformer-based recommendation system which is designed by Qiwei Chen and his team in research paper [Behavior Sequence Transformer for E-commerce Recommendation in Alibaba](). This model leverages the sequential behavior of the users in watching and rating movies, as well as user profile and movie features, to predict the rating of the user to a target movie.

BST model aims to predict the rating of a target movie by accepting the following inputs: A fixed-length sequence of movie_ids watched by a user; A fixed-length sequence of the ratings for the movies watched by a user; A set of user features, including user_id, sex, occupation, and age_group; A set of genres for each movie in the input sequence and the target movie; and A target_movie_id for which to predict the rating. And BST model aims to predict the rating of a target movie by accepting the following inputs Incorporate the movie features into the processing of the embedding to each movie of the input sequence and the target movie, rather than treating them as "other features" outside the transformer layer. Utilize the ratings of movies in the input sequence, along with their positions in the sequence, to update them before feeding them into the self-attention layer. Finally this model achieved a MSE at or around 0.7 on the test data.

Summary

In conclusion, this project attempted to test all three models in order to see how efficient and accurate the models can be for our data recommendation system. The Content-based Model and Collaborative-filter Model both have a rather low MSE in our test(0.04) compared to that of the BST model(0.70). Nevertheless, the difference in MSE could potentially be the result of low epoch times. The Content-Based Model is the most stable one since it mainly focuses on the user's input while its simplicity has its own limitations. The Collaborative-filter model ,on the other hand, solves that problem by adding a feature which focuses on similarity in preference between users, thus enhancing the accuracy of movies the system recommended. The BST model still has a large potential for future research to develop. It evolved from earlier two models and leverages both the sequential user's daily watching behavior and the user profile and movie features, to predict the rating of the user to a target movie.

Reference

https://movielens.org/

https://github.com/lyhhhhhh1006/DL-finalproject/blob/6405bd9c5bb2d96367986d55efc9ca6d01f419e1/codes/content-based_model/data_processing.ipynb

https://medium.com/swlh/recommendation-system-for-movies-movielens-grouplens-171d30be334e

https://arxiv.org/pdf/1905.06874.pdf

https://towardsdatascience.com/the-4-recommendation-engines-that-can-predict-your-movie-tastes-109dc4e10c52

https://dl.acm.org/doi/pdf/10.1145/371920.372071

Github

https://github.com/lyhhhhhh1006/DL-finalproject