Anly 590 - Project Proposal

Team member: Jiawei Yu, Yuqi Wang, Dantong Chen

1. Project's Goal and Objectives

a. The problem we want to solve

People sometimes cannot find appropriate movies they need, and it's time consuming to search by themself. Therefore, we want to build a **movie recommendation system** which could recommend appropriate movies to specific users.

b. The expected end result

As we specify a user as input, we will get **5 recommended movies** whose possibilities of being highly rated by this user are highest.

c. Unique points

Our movie recommend system considers three situations and contain all types of users:

i. For users who don't have any rating history, the system will recommend the most popular movies to them.

ii. For users who have rated less than 30 movies, the system will recommend movies which have highly rated from the other similar users to them.

iii. For users who have rated equal to or greater than 30 movies, the system will predict the users' rating on the new movie based on their previous ratings, and recommend them with potential high ratings movies.

We will use deep learning to predict the users' ratings and compute similarity of different users for recommendation. It means in our model, we will combine two methods.

2. Data

a. There is a stable benchmark dataset from grouplens.org, called MovieLens 20M Dataset. Here is the link of the reference dataset: https://grouplens.org/datasets/movielens/20m/

b. This dataset describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 20,000,263 ratings and 465,564 tag applications across 27,278 movies. These data were created by 138,493 users between January 09, 1995 and March 31, 2015.

c. Users were selected at random for inclusion. All selected users had rated at least 20 movies.

d. This dataset is a combination of six csv files: ratings.csv, tags.csv, movies.csv, links.csv, genome-scores.csv and genome-tags.csv. The following table shows all of features in each csv file:

File	Features
ratings.csv	userId movieId rating timestamp
tags.csv	userId movieId tag timestamp
movies.csv	movieId title genres [d]
links.csv	movieId imdbId tmdbId
genome-scores.csv	movieId tagId relevance
genome-tages.csv	tagId tag

e. Genres:

   There are 19 genres of movies as below:

   i. Action
   ii. Adventure
   iii. Animation
   iv. Children's
   v. Comedy
   vi. Crime
   vii. Documentary
   viii. Drama
   ix. Fantasy
   x. Film-Noir
   xi. Horror
   xii. Musical
   xiii. Mystery
   xiv. Romance
   xv. Sci-Fi
   xvi. Thriller
   xvii. War
   xviii. Western
   xix. (no genres listed)

f. Possible limitations of the data

   This dataset only includes users who had rated at least 20 movies. So it may be hard for our model to recommend movies correctly for users who had rated less than 20 movies. Also, this dataset does not include the latest movies so we cannot recommend new movies to our users.

g. Areas where you could improve the collection of future data

   We can also collect which movie pages each user has clicked recently and the ratio of rating by clicks.

h. Why the current data source is appropriate

   MovieLens is a web-based recommendation system that recommends movies for its users to watch. The record is well rounded and authoritative. Moreover, There are many types of research conducted based on the MovieLens data sets.

i. If your team is working on a supervised problem, what is your input feature vector and output?

   The input feature vector consist of a user, a movie, the movie genre, and all genome scores of the movie. The output is rating prediction of this movie by this user. It may be revised later while actually working on this neural network.

3. **Assessment Metrics**

Write a short outline describing:

a. What loss metric you will be using and why?
   i. We will use mean squared error as loss metric because it’s a regression problem which predict the specific user rating to the specific movie (5-star based rating).
   ii. In addition, we will check if the recommendations are appropriate enough by listing a specific user’s rating history and a recommendation list for him or her since the accuracy of a recommendation does not have an absolute numeric value. We can only measure this by human.

b. What baseline datasets will you be using to evaluate your models performance?

   Same dataset. We can split it into training set and testing set.

c. What other models are used as baselines - how do you expect your approach to compare?

   The baseline model we will use is collaborative filtering model for calculating the similarity and the deep neural network for the rating prediction.

d. What is considered **state of the art** in the field and how does it compare to your method?

   There are many video websites and companies are working on movie recommendation. Some of them focus on the movie part and compare the similarity of the movie. They are using singular value decomposition to do matrix factorization to calculate similarity. We could also use this method to calculate the similarity of the users.

4. Approach

Write a paragraph for each of the questions below:

a. At a high level what are the expected outcomes of your project

– What approach will you be taking and why?

– Describe possible limitations of your approach

We will use two method, User-User Collaborative Filtering algorithm and RNN in the project to compare which model is better. User-User Collaborative Filtering algorithm and RNN are widely used in recommendation system field. RNN is suitable for modelling sequential data. Unlike feedforward neural network, there are loops and memories in RNN to remember former computations.

However, the limitation of User-User Collaborative Filtering algorithm is sparse matrix of user-item pair. Due to recommendation system need a large number of rating history to make prediction, the result will bring great error since most user do not have enough record to match similar users. On the other hand, traditional RNN is difficult to learn long-dependency correlations in a sequence, and neural network may introduce additional hyperparameters in some cases.

b. Where are you going to train your model?

– Will you be using a cloud provider or running it locally?

– What are some limitations if running it on the cloud vs locally?

We will train the model locally. Since the dataset is huge, we could choose sample randomly to train the model. Local running limits the size of the training data and has high time cost. Cloudy computing is less controllable over the function and execution of services within cloud-hosted infrastructure. Moreover, Adopting cloud solutions on a small scale and for short-term projects can be perceived as being expensive.

c. What API will you use to train your model (i.e. Tensorflow, Keras, PyTorch)

In this Project, we will use Keras, scikit-learn to train the model.