The Proposal of a Recommendation System for Movielens

LI Yunli, LAU Kwan Yuen, LAI Wendi, WANG Xueying, YANG Rongfeng
HKUST MSBD 5001 FMVP Team

1 Introduction

Our team proposes to build a recommendation system using dataset provided by Movielens. Based on the dataset, we train a model to predict movies that the user prefers by analysing the previous data and then we recommend them to the user. We also plan to build a web interface to demonstrate our work.

The document is organized as follows: we first discuss the dataset we use in the (§ 2). Then at the (§ 3), we describe problem we are exploring with our dataset and also the possible solution. Finally, we show the expected outcome of our project (§ 4) and the scheme & distribution (§ 5).

2 Dataset

We use the dataset `ml-20m` which is provided by the Movielens to do our project. The dataset describes 5-star rating and free-text tagging activity. It contains 20000263 ratings and 465564 tag applications across 27278 movies. These data were created by 138493 users between January 09, 1995 and March 31, 2015. This dataset was generated on March 31, 2015, and updated on October 17, 2016 to update `links.csv` and add `genome-scores`, `genome-tags` files.

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided. The data are contained in six files,

- **genome-scores.csv** A data structure that contains tag relevance scores for movies which is described as a dense matrix with value's range [0, 1]. The tag genome encodes how strongly movies exhibit particular properties represented by tags (atmospheric, thought-provoking, realistic, etc.). Data format: movieId, tagId, relevance.
- **genome-tags.csv** Containing the tag descriptions for the tag IDs in the genome-scores.csv file. Data format: tagId, tag.
- **links.csv** Containing identifiers that can be used to link to other sources of movie data such as imdb and themoviedb. Data format: movieId, imdbId, tmdbId.
- **movies.csv** Contains the movie information. Each line of this file after the header row represents one movie. Data format: movieId, title, genres.
- **rating.csv** Containing rating data by users. Each line of this file after the header row represents one rating of one movie by one user. Data format: userId, movieId, rating, timestamp.
- **tags.csv** Containing the tags information that users use to describe movies. Each tag is typically a single word or short phrase. Each line of this file after the header row represents one tag applied to one movie by one user. Data format: userId, movieId, tag, timestamp.

3 Problem

3.1 Overall Description

The core problem we explore is selecting several films for the users according to the movies they have seen. We notice that the genome-scores.csv dataset is very useful which has helped us extract 1128 features (the file call it *tags*) from the user-contributed content including tags, ratings, and textual reviews. Each movie can be descibed as a feature vector. It's a kind of dense matrix like:

$$scores = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{bmatrix}$$

$$\text{where } n = \text{feature number},\ m = \text{movies number}$$

We can use it to train a model to find the movies with close relevance. We also plan to do PCA (Principal Component

Analysis) for the dataset so we can extract the key features further and clear away distractions.

However, the features and relevance are computed by the Movielens machine learning algorithm. We tend to use it but not only rely on it. We tend to do our own features extraction on other dataset such as movies.csv, rating.csv, tags.csv. It is worth mentioning that we are likely to use spider to capture the information in imdb and themoviedb website through links.csv in order to provide more data for us to analyse. If we have to process users textual reviews, we need NLP knowledge.

3.2 Solution Framework

To address the aforementioned problem, we propose a solution framework with two independent modules, i.e. recalling module and ranking module. When recommending movies to a particular visitor, recalling module first filters out a majority of less relevant movies from the candidate list. After that, ranking module will sort out movies with high rank from the remaining candidates for the visitor. In practice, we also need to solve the user cold start problem.

In recalling module, we apply a user-based collaborative filtering approach to obtain the input of ranking module. Specifically, we first find out users, who have similar preferences about movie to the visitor's, based on a rating matrix calculated from previous rating records. After that, we can obtain a list of movies, which those users having similar preferences have watched before, as the input of ranking module.

In ranking module, we propose two different solutions. For the first solution, we first calculate the average genome scores over movies which the visitor has watched before, and denote the average genome scores as S. Then, we calculate euclidean distance from S for each candidate, and add the weighted distance to the weighted corresponding overall rating as the rank for a given candidate. Finally, we sort out movies with high rank for the visitor. For the second solution, we treat movies which the visitor has watched before as user features, and utilize those features to train a regression model. This model, taking features of a particular user and genome scores of a movie as inputs, can predict the rating of a movie given by a particular user. With this model, we can easily recommend movies with high predicted rating to the visitor.

However, in case of recommending movies to a new visitor, we are unable to perform both recalling and ranking due to the lack of previous rating records. Considering this problem, a.k.a user cold start problem, there is a naive solution, i.e. to simply recommend movies with high overall rating to new visitors.

4 Outcome

The outcome contains models and a website.

- **Models**. Make Several practicable models and an ideal one to predict movies that the user prefer.

- **Website**. Build a website recommendation system as an interface to interact with users. In the website, users can choose their favorite movies and it'll recommend some other new movies to the users.

5 Scheme and Work Distribution

Our project timetable is in Table 1:

Table 1: Project Scheme

Deadline	Works
Oct. 14th	Dataset analysis and model discussion
Oct. 21th	Ensure the solution
Oct. 28th	Coding
Nov. 4th	Coding
Nov. 11th	Prepare PPT and presentation. Finish final report
Nov. 18th - 25th	Do presentation
Dec. 2th	Submit a final report

Our work distribution:

YANG Rongfeng Project managing, frameworks, coding.

LAI Wendi Dataset processing, coding.

LAU Kwan Yuen Ideal solutions analysis, frameworks, coding.

WANG Xueying PPT, report, coding.

LI Yunli Literature translation, presentation, coding.