Proposal: Movie Recommender System for IMDB and Douban

Abstract

IMDB and Douban represent two of the largest scale movie databases and social networking service. Previous works mainly focus on the movie recommendation for English-speaking users using IMDB dataset due to the shortage of available datasets. In this project, we describe the movie recommender system for IMDB and Douban with millions of ratings and thousands of movies and narrow down the gap between them, considering the volume and variety properties of big data. The recommender system composes two stages: candidate generation and movie ranking. We discuss how deep learning bring performance improvement when considering temporal information. The system provides personalized recommendations and will benefit millions of movie lovers.

1 Motivation and Background

The Internet has brought a lot of conveniences and a lot of fun to our lives nowadays for it can provide a large quantity of relevant information according to your input search keywords. However, how to get more precise data you want in this huge interconnection is a problem. When you want to watch a movie, how to choose on that website, what type of movie to watch, watch the latest movie, or watch a classic old movie need to be considered together.

The recommender system is one of the most effective ways to solve the above problem. The recommendation algorithm is a fitting function for content satisfaction, involving user features and content features as two sources of dimensions required for model training and rate or comment can be used as a quantified Y value, so that you can perform feature engineering, construct a data set, and then select a suitable supervised learning algorithm to train. After obtaining the model, recommend the preferred content for the customer. The big amount of data available from IMDB and the recent web crawled data from Douban enable us to do data analytics in the scope of the recommender system.

2 Related Course Topics

This project is related to MapReduce (ch.2), clustering and classification (ch.7), recommender system (ch.9) and machine learning (ch.9&10). Also, some text processing procedures in NLP will be involved to implement features extraction from the dataset.

3 Deliverables

We plan to submit a recommendation application with a report describing the detailed design process and experimental performance. The measurable items include the recommendation accuracy (RMSE), the time to generate recommendations, etc.

4 Dataset Overview

In this project, we focus on the recommendation of movies and related data analytics. Unlike previous research or applications, we not only aim at IMDB users but also Chinese Douban users.

The IMDB has its built-in database interface, which contains the basic information and average ratings for each movie. However, it doesn't contain the rating by every user, which means we can't trace back to the movie-user interaction. Though not able to use directly, it contains some helpful information for potential analytics, e.g. the directors, where people tend to watch movies of their favorite directors. Below is an example of the movie information provided.

Movie Title Basics

Table 1: Movie Information from IMDB interface

Titile ID	Primary Title	Average Rating	Num Votes	...
tt0000001	Carmencita	5.6	1537	...
tt0000002	Le clown et ses chiens	6.1	185	...

Thanks to the MovieLens [1], who enable us to do data mining on a large dataset, which contains the movie information and the ratings by users for IMDB and TMBD. The dataset recommended for education and development purpose records information about 58,000 movies and corresponding 27,000,000 ratings by 280,000 users. The collected information includes each movie's unique id, start year, genres, title, tags, and links. They also provided us with the timestamp of the event when users comment and rate the movies. Below we give several examples of the datasets and information we will extract for our proposed methods.

Movies

Table 2: Movie Information from MovieLens

Movie ID	Title	Genre	...
12	Dracula: Dead and Loving It (1995)	Comedy|Horror	...
30	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)	Crime|Drama	...

Ratings

Table 3: Rating Log from MovieLens

User ID	Movie ID	Rating	Timestamp	...
1	231	5	96498179	...
322	30749	4.5	12176777759	...

One problem with this dataset is that even though we have the movie id in MovieLens system, it doesn't match the movie id from the IMDB interface. So, when combining these two datasets to extract more meaningful information, we need to preprocess and clean the data first. From the examples above we can see the only way to combine them is through title names.

Recent research concerns the disconnect between the size of academic data sets and the scale of industrial production systems [2]. [2] proposes to generate more massive user/item interaction data sets by expanding the MovieLens dataset to bridge that gap. The synthetic dataset contains 1 billion user-movie interactions and is about 200GB.

On the other hand, we also collect the data from Douban, which is a famous Chinese social networking service including movie ratings. One recent web crawling dataset has been constructed for research purposes [3]. The dataset contains about 140,000 movies with corresponding 4 million ratings by 630,000 users. The collected information contains movie id, title, genres, regions, tags, average rating, start year, etc. Below we give several examples of the files in this dataset.

Movies

Table 4: Movie Information from Douban

Movie ID	Title	Directors	Genres	Year	...
26670818	情定河州	尹哲	剧情/爱情	2049	...
1307315	哪吒闹海	严定宪/王树忱/徐景达	动画/奇幻/冒险	1979	...

Ratings

Table 5: Rating Log from Douban

User MD5	Movie ID	Rating	Rating Time	...
0ab7e3efacd56983f16503572d2b9915	5113101	2	2018-09-05 19:42:07	...
c98155000a3420758910ac19414b7118	4824996	5	2012-07-25 22:03:32	...

5 Methodology

The methods for the recommender system can be usually categorized into content-based recommendation systems, collaborative filtering and more recent deep learning-based methods [4]. While conventional collaborative filtering methods (e.g. singular value decomposition) can achieve reasonable prediction results, they are not able to predict very accurately when concerning more factors such as viewing sequence. In this project, we propose to use a two-step hybrid approach to prepare a movie recommendation list. In the first stage, we narrow the candidate movies down to several hundred by rule-based methods to fulfill basic requirements for each user. In the second stage, the candidate movies are further ranked in the order based on a regression neural network.

The main techniques and algorithms we will apply includes MapReduce (Spark), clustering, wording embedding, deep learning.

5.1 Data Preprocessing

As mentioned above, one challenge of this project is to construct an integrated and clean dataset for further processing. Though the MovieLens and Douban datasets contain big amounts of data, there exist many noises and outliers. The first step is to extract useful information from separate files and combine them into one tabular file. We find that in the Douban data, particularly, a reasonable portion of movies have no ratings. On the other hand, due to some reasons, some movies haven't shown up in mainland China and have fake publish years (e.g. year 2049 in table 4). We propose to use Hadoop MapReduce or Spark to clean up and integrate the datasets because of the large amount of data.

One important thing before feed the movie ratings to the model, is to embed the movie into some numerical representation, e.g. embed "Dracula: Dead and Loving It" into [-3.6, 2.4, 1.7]. One simple way is to use a one-hot encoding. However, the one-hot encoding can't represent the contents or genres of the movie, while occupying a lot of memory. Another way is to use Word2Vec for each word in the movie title and genres to get feature vectors followed by a weighted average of them to build the final representation for each movie [5]. We borrow a similar idea of word embedding from NLP and take care of the sequence information. We regard movies as tokens and user view history as sentences. In practice, one way to handle new movies is to use an average of similar genres.

We can also inspect the result of word embedding. Quantitatively, we can visualize the word embedding for movies with t-SNE. We hypothesize that after embedding, movies will be close to movies with similar genres. We can also compute the distance between different word embeddings in the same genre group or different genre group and check if the distance meets our expectations.

5.2 Recommendation

In this project, we want to provide personalized recommendations for each user, considering accuracy, variety, and efficiency. The recommender system composes two stages: candidate generation and movie ranking.

The first step of our recommender system is a rule-based generation process. There are tons of movies in our database. It's not efficient to feed all of them to the model for one specific user. So, in the candidate generation stage, the large amount available movies are narrowed down to several hundred. The requirements include specific genres, related directors, movie regions, excluding older movies, etc.

We treat the movie recommendation problem as a regression problem. The idea is to train a model to predict the rating score if one movie is recommended to a user given the user rating history. The rating score ranges from 1 to 5 according to the MovieLens and Douban dataset. Considering the taste of a user may change over time, we want to also capture the temporal information when generating recommendations. Having the word embedding of each movie and the rating history of the user, we feed the feature vectors into RNN/LSTM layers in a sequence manner.

Suppose the movie rating history of a user *M1*, *M2*, ..., *Mn* with corresponding rating scores *R1*, *R2*, ..., *Rn*. For simplicity, let *Mi* denotes as the feature vector for the movie *i*. The input for the model is a sequence of feature vectors, e.g. *M1*, *M2*, *M3*, *M4*. We expect the model to predict the rating score for the movie after seeing the movie sequence. Table 5 is a more specific illustration.

To get the prediction of the user for one movie, we need to consider both the feature vector of the movie and the user. The movie feature vector is the output of the RNN layer, while we can treat the average of movie vectors the user viewed as the token for the user. Besides, other information such as the time since the last watch on a genre, etc.

Table 5: illustration of the input to the model

Input	Label
M1, *M2*, *M3*, *M4*	*R4*
M3, *M4*, *M5*, *M6*	*R6*

During inference time, the users are supposed to give ranking scores to some movies. The initial recommendation relies on the requirement fulfillment methods, e.g. genre. Then based on the provided information, we rank the movies according to the prediction score by the recommendation model. As we collect user's more ranking history, the sequence relationship affects. Meanwhile, we also consider the diversity of recommendations according to the long tail strategy.

In addition to the recommendation, we will conduct some data analytics, including the

clustering of the movie, the top words related to the movie via Twitter, etc.

6 Demonstration

Our main function in this project is the movie recommender system. The demonstration process is as follows:

1) A new user needs to choose the dataset (IMDB or Douban);
2) The user needs to choose his/her favorite genres or rate some movies;
3) The recommender system will generate a few movies based on the user's preferences.
4) As the user rate more movies, the recommendations will change over time.

Besides the recommender system, since we have the genres and comments (the Douban dataset) information, we will conduct the sentiment analysis and draw the histogram of top keywords. If possible, we will trace the change of popular words for a movie using streaming techniques via Twitter API.

7 Existing Work

We explored some papers and discovered several common strategies which have been used to build a recommendation system. Some systems recommend movies which are similar to users' previous interest [1] or which are preferred by other users who share close interests [2]. These works use collaborative filtering which models the people's tastes and recommends movies with similar characteristics to the group of users with the corresponding preferences using MinHash Clustering or neural networks. Other works tend to suggest movies which are least known to the user [3] or give high possibilities to those movies which are liked by the majority of users [4] and they are defined as prediction preference based on popularity and diversity.

The main difference between previous work and our work is the consideration of temporal information, where the previous work treats the recommendation as a static process. On the other hand, while the classic collaborative filtering can give reasonable recommendations, we hypothesize that deep learning-based method can boost the performance by a large margin. Till now, our project is the first attempt to analyze the big amount of data for Douban database.

8 Timeline

Before 30/09: Identify the project, prepare the dataset.
Week 5 – week 6: clean up and prepare the dataset for model training.
Week 7 – week 9: recommendation model design and experiments.
Week 10 – week 12: model tuning, data analytics, demo preparation.
Week 13: final report.

References

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19.

[2] Belletti, F., Lakshmanan, K., Krichene, W., Chen, Y.F. and Anderson, J., 2019. Scalable realistic recommendation datasets through fractal expansions. arXiv preprint arXiv:1901.08910.

[3] https://www.csuldw.com/2019/09/08/2019-09-08-moviedata-10m/

[4] Covington, P., Adams, J. and Sargin, E., 2016, September. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM conference on recommender systems (pp. 191-198). ACM.

[5] Yuzi, L., Yue, L., Yawen. S., 2019. Recommender System for Publisher of Technology News. http://web.stanford.edu/class/cs341/project/Luo-Li-Sun_report.pdf

[6] Yang, Qin. "A novel recommendation system based on semantics and context awareness." Computing 100.8 (2018): 809-823.

[7] Chen, Jianrui, et al. "Personal recommender system based on user interest community in social network model." Physica A: Statistical Mechanics and its Applications 526 (2019): 120961.

[8] Paul Resnick, R Kelly Garrett, Travis Kriplean, Sean A Munson, and Natalie Jomini Stroud. 2013. Bursting your (filter) bubble: strategies for promoting diverse exposure. In Proceedings of the 2013 conference on Computer supported cooperative work companion. ACM, 95–100.

[9] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-ranking. In Florida AI Research Symposium (FLAIRS). ACM, to appear.