Designing and Developing a Personalised Recommender System

Thomas Butterfield
dept. Computer Science
Durham University
Durham, United Kingdom
thomas.butterfield@durham.ac.uk

I. Introduction

A. Domain of application

The domain of this application is a recommender system for bars, using the Yelp dataset. Decide whether I want to define the time-span for the date, depends on if there is too much data from all available time period. I am also taking into accounts Covid-19 data when generating recommendations.

B. Related work review

Some related work

C. Purpose/Aim

The purpose of this application is to give suitable suggestions for a user to go to. Using information about the user such as their location and personal preferences.

II. Methods

A. Data description

The data I am using is taken from the Yelp dataset, it includes user reviews of different businesses and services in a specific location. There are 10 cities included in the dataset: Montreal, Calgary, Toronto, Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland. These user reviews include ratings, text feedback, and other such information.

B. Data preparation and feature selection

Prepare data Of the 209,393 businesses in the dataset, 168,903 are still open. There are 43,965 restaurants which are open and contain 'Restaurants' as one of the categories. I then picked the top 12 most popular categories of restaurant: American, Mexican, Italian, Chinese, Seafood, Japanese, Canadian, Mediterranean, Indian, Thai, Middle Eastern, Vietnamese (in that order). When filtered for restaurants which fall into at least one of these categories, 20,565 had at least one check-in in 2019, but 333 of these were temporarily closed due to covid but should be open now, and 1 is still closed until June 2021. 19,862 restaurants had a review written about them in 2019, there were 503,979 reviews written in 2019 about these restaurants. There are 2,956,316 reviews of these restaurants. 481,073 tips for these restaurants. 1,968,703 users. None of these restaurants are temporarily closed according to the covid features data. 20,573 restaurants listed with covid features, 4 restaurants which make up 8 duplicate entries (4 have 3 entries), the only difference between their entries being in the "Virtual Services Offered" section. Allow to pick user ID of 10 users with most reviews made, to avoid cold start problem 4 users have over 10,000 reviews, the most is 14,455 (Victor, 8k3aO-mPeyhbR5HUucA5aA). I actually want to select the most reviews of the places I am accepting, restaurants. When taking the most reviews of restaurants in teh dataset, a threshold of 600 reviews gives a list of 10 users, with the most being 1747. I then checked each of these for how many different restaurants they had reviewed and the greatest number of times they had reviewed a single restaurant to check that none of them were just users who reviewed a small number of restaurants a huge amount of times each. All 10 users passed this criteria, with even the least active user having reviewed 496 unique restaurants, which will certainly avoid any cold-start problems. I split the businesses into 10 groups, by the city which they are closest to, I did this by using latitude and longitude rather than State or City since this does not give actual distance and restaurants close to a border could then be mis-categorised. Since there are 4,223,201 reviews of restaurants I filtered them by removing any which had zero 'useful' votes. When filtered to reviews with at least 1 'useful' vote, there are 1,735,732 and when at least 2 votes, 836,929 At least 3 votes, 462,412, at least 5 votes 190,902. 51,737 with at least 10 votes

Top categories: American, Mexican, Italian, Chinese, Seafood, Japanese, Canadian, Mediterranean, Indian, Thai, Middle Eastern, Vietnamese

The entire Yelp dataset is huge and much of it is not necessary for my domain of bars. As such I prepared the data by eliminating any data not relevant to my domain. I selected features such as user ratings and a particular user's average rating, since if a user typically rates places they go highly, but rates a particular bar low, this is more significant than a user who always rates places low.

C. Hybrid scheme

Which two algorithms A hybrid scheme is a good way to design a recommender system, since you can get the best of both algorithms if done properly. I am using a Cascade Hybrid Recommender System, since this has been shown to accurate predictions in my domain of restaurant recommendations [1].

The two systems are Collaborative Filtering and Content Based.

Meaning better recommendations than either algorithm could achieve individually. Cascade with knowledge-based and collaborative

D. *Recommendation techniques/algorithms*

The first recommender system is collaborative filtering. I am using item-based collaborative filtering, since the number of users is larger than the number of items (1,968,703 users vs 209,393 businesses in the entire dataset), this gives greater accuracy of predictions. In this case the list of available items does not change, so a user-based method would be unnecessary. One of the requirements for this system is justifiability, and item-based methods make it much easier to explain why certain predictions were made. Item-item collaborative filtering looks for restaurants that are similar, in terms of how people rate them, to the restaurants that the user has already rated and recommend the most similar restaurants. Filtering reviews by city and then using CF won't work because the active user may not have ever reviewed a restaurant in that city, so there is no way to assess similar restaurants to ones they have already rated. Instead I carried out CF on all high quality reviews and produce an ordering of recommendations, which I then filter by city.

I am using a weighted mixed combination of these two systems in order to produce the results of my hybrid recommender system.

E. *Evaluation methods*

How to evaluate

III. Implementation

A. *Input interface*

The system has a command line based interface, so all user input of the input is given via the command line. The system recognises the active user by their unique user ID, some example IDs are provided in the README.txt file for testing and demonstration purposes. Only explicit user data is gathered in order to make the system more explainable and transparent.

The input interface is the command line. The program offers users opportunities to input information about themselves, as well as make choices from a selection of items which the system provides or suggests.

B. *Recommendation algorithm*

What algorithm

C. *Output interface*

The output interface is the command line. The system can output recommendations for bars which the user might like, as well as information about how the system works and why certain suggestions were made, at the user's request.

IV. Evaluation results

A. *Comparison against baseline implementation*

Compare vs generic suggestions

B. *Comparison against hybrid recommenders in related studies*

Read some papers

C. *Ethical issues*

People's personal data

V. Conclusion

A. *Limitations*

What can it not do?

B. *Further developments*

What could I do in the future [2].

References

[1] R. Burke, "Hybrid web recommender systems," *The adaptive web*, pp. 377–408, 2007.

[2] L. Martinez, R. M. Rodriguez, and M. Espinilla, "Reja: A georeferenced hybrid recommender system for restaurants," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3, 2009, pp. 187–190.