Recommender Systems for Analysing Yelp Dataset

Alasdair Cooper
Department of Computer Science
Durham University
Durham, United Kingdom
ttnt59@durham.ac.uk

Abstract— **This is a report into the use of two different recommender systems for analyzing a part of the Yelp dataset.**

I. Introduction

The aim of this report, and the development of the recommender systems is to identify the best recommender system suitable for the dataset. The dataset in question is the Yelp dataset [1]. A subset of their existing database containing businesses and reviews. Only the JSON version will be accessed and used, as there is also an image dataset available.

The domain of this dataset are user reviews of businesses across 8 metropolitan areas. There are around 8.6 million reviews of just over 150 thousand businesses. This means the dataset is incredibly large. The business data alone has a file size of 100MB while the user review data is over 5GB.

II. Methods

A. Data description

The dataset contains data about businesses, users, reviews, and other categories that will not be used.

Due to the enormous size of the user review data, it will also not be used. Custom data will be generated instead to simulate user interest in different businesses. The size means it could not be loaded into my machine unless using a special program for streaming data from files. Although it would have still been able to be processed, it would take an incredibly long time. Thus, it was not very suitable for the implementation.

The business data contains information about the position of the business, when it opens, and certain information about the categories applied to it, such as “burgers”.

The custom user data generated includes weighted values for how much they liked a specific business. As this data is only needed for collaborative filtering, it is generated as a matrix not a labelled dataframe.

B. Data preparation and feature selection

The data is sampled to a fixed number of businesses, and some features will be removed. This included positional data, the latitude and longitude of the business. This data could be useful as a key part of a user visiting a business is the proximity of the business, However, the data in question is spread across the USA and Canada in such a way that the entirety of the data would have to be sorted to even produce a few locations within 10km or so of the user (assuming the user’s position is somewhere in North America). This would not lead to a prompt user experience.

The dataset contains categories for each business. These are tags that associate the business with some keywords, for example “Shopping” or “Automotive”. These were extracted into their own columns with a value of “1” meaning they were present in the business data item and a value of “0” meaning they were not. This resulted in data formed of over 500 columns, but it also meant the data would be more ready for processing later on. Certain useful and interesting attributes were chosen to be extracted. This included information about restaurant delivery or takeaway, disabled access, and whether they allowed dogs. The most important attribute is most likely disabled access, and it is also the most prevalent, as many businesses do not serve food. The star rating of each business was kept forming the basis for calculating the accuracy of ratings. Informational data including the name and address was also preserved as well as the opening hours. All other data was discarded.

C. Recommendation techniques and algorithms

The first chosen recommender system method was Content Based Filtering (CBF). It was chosen as the data was structured by features in a vector structure. New items could be easily recommended based on what the user had previously chosen. CBF does not require anything extra on top of the user ratings for each item. Although there is extra data available, it is not in a suitable format for easily using for recommendations and would require extensive processing hindering the user experience. The CBF data was weighted using the TF/IDF technique and normalised using the cosine method. This gives the predictions greater focus on more common categories such as “Food” and “Shopping” as opposed to “Reflexology”, thus presenting more useful recommendations.

The second chosen recommender system was Collaborative Filtering (CF). It was chosen as user data could be easily generated for the items, and randomly weighted to indicate user preferences. It does not require content; however we already have that anyway, but it does require ratings. As mentioned, these ratings could be generated, or they could be taken from the user review dataset. Unfortunately, due to the size and complexity of this dataset, this is not a good fit for this specific recommender system. The algorithm used for the CF model fitting and recommending was Alternating Least Squares, and algorithm based on the paper “Collaborative Filtering for Implicit Feedback Datasets”[2]. This algorithm is a model optimised for implicit feedback recommenders and runs well compared with similar recommenders.

D. Evaluation methods

The primary measure of accuracy chosen was Root Mean Squared Error (RMSE) as these recommenders’ produced ratings predictions, and the only suitable true rating is the star rating for each business. The predicted rating was the prediction calculation for the CBF and the normalised scores for each business for the CF. Both values were normalised to the same range as the stars, which was also normalised to between 0 and 1.

The second evaluation metric was coverage, as the necessary information is available for the size of the sample and the number of items the user has viewed.

III. Implementation

A. Input Interface

The user can enter a name to identify themselves, as well as use the same name later on to continue with the same data. They can then choose to either browse or be recommended some data items. If they choose to browse, they are presented with up to five different items, as well as a limited amount of data on each item. They may select one item and view the full data on it and may proceed to opt to like or dislike it. They are then presented with five new items with which they can repeat the process or return to the original menu. There is also a help option from the original menu that informs the user of the functions of "BROWSE" and "RECOMMENDATIONS". Selecting of items is done numerically in the bottom line of the console. Text is presented in a box preserved between console updates using curses, a Python module.

B. Output Interface

If the user chooses the recommendations option, they can then further choose the recommender system type. They are then presented with the top 5 ranked recommended items.

IV. Evaluation

Defining characteristic	Content based RS RMSE	Collaborative Filtering RS RMSE
Food	0.45924574186796446	0.7131868915782436
Food	0.5401432406630764	0.7532243615122325
Shopping	0.4897562341706503	0.7761947967344893
Shopping	0.2800840445644245	0.685222776729749
Neither food nor shopping	0.5654714334168444	0.7099295739719539

Both recommender systems show a medium-high RMSE. However, CBF performs quite a bit better than CF in this area. This is mostly likely due to the recommendation of more user-oriented results. The recommendations will be heavily personalised and not as vague as the CF recommendations. The CF recommendations recommend similar users' recommendations, but they are clearly not as well fitted to the current user.

The coverage of each recommender is full of all the items in the sample, including ones the users already confirmed they liked or disliked. The coverage of all items in the dataset is poor. The coverage for the entire dataset is about 0.0013 with the default sample of 200.

V. Conclusion

Although the CBF performed better, the CF successfully provided more diverse content, as is expected. Many changes could be made to improve the systems, and a few can be made to improve the data samples.

Primarily, there were many more features that can be sampled. The attributes (for example whether dogs were allowed) could be sampled fully, like the categories were. Furthermore, with extensive pre-processing and optimisation the location data could be calculated per user for small samples at a time, perhaps with some optimisation from threading to not impact the user experience too much. The data could be grouped for metropolitan areas and the user location could be simulated as being in, or close to that area. This would result in better recommendations as most likely the user will prefer closer business to ones further away. Furthermore, the data could be sampled in full with some pre-processing. The enormous number of categories across 160,000 businesses meant the categories had to be extracted after the sample but this was incredibly time consuming and produced an enormous number of columns that meant processing later on was slow as well, impacting the user experience. This could have been handled by filtering out extremely low frequency categories.

The CBF could not have been improved much. It performed well and gave accurate predictions. A hybrid model with CF might have presented more diverse recommendations to the user.

The CF could have been improved by running with different algorithms, or by improving the quality of the data of other users. The calculated data could be structured instead of random so that places with similar categories were linked in ratings. Alternatively, the user review data could be pre-processed over a period of time into small samples of users with accurate ratings for items. Overall, the CF is probably the least accurate of the two due to the randomness of the user data with regards to the actual data. Improvements of the CF would lead to much lower RMSE.

VI. References

VII. References

[1] Yelp, "Yelp Open Dataset".

[2] Y. K. C. V. Yifan Hu, "Collaborative Filtering for Implicit Feedback Datasets," in *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, 2008.