Restaurant Recommendation System

Al Baraa Abd Aldaim, Mehmet Basaran, Yousef Qaddura
Github: https://github.com/mbasaran97/Food-Rec-Erdos23

Overview

Recommendation problems are prevalent in many areas of business such as recommending movies or entertainment (e.g. Netflix), food or restaurants (e.g. UberEats or Yelp), or products in a marketplace like Amazon. As these recommendation algorithms drive sales, they are also of great relevance to advertisers. At the same time, users are becoming more conscious about privacy issues and generally prefer to share less identifying information about themselves, which makes the recommendation problem more difficult.

We worked on implementing a recommender system that uses the Yelp Dataset to suggest new restaurants to the user by using only their numerical ratings of restaurants, without any identifying information. Our model performed better than the baseline of recommending restaurants based on their average rating, which is itself a quite robust system, showing that we can glean some preferences just from the numerical ratings.

Stakeholders: users, restaurants/product providers, advertisers, online marketplaces, delivery platforms

KPI: the RMSE of the predicted numerical ratings

Approach

Matrix Completion Step: The first step in our approach is predicting the unknown ratings of known users in our dataset. We use a Latent Factor Recommender Systems (LFRS) factorization of the normalized user-restaurant rating matrix combined in a mixture with the average ratings of each restaurant.

Prediction Step: The second step is to dynamically predict a new user's restaurant ratings given only a few. We compared three approaches. The first is finding nearest neighbors to the user. The second is a weighted average based on distance to other users. The last approach was to use a variant of the LFRS factorization approach where the restaurant factors are fixed from the first step and the user factors are trained.

Results

Our cross-validation results showed that the model where we use normalized LFRS for completion and train user factors for prediction consistently outperformed the baseline averaging model.

In particular, with the weighted average step giving 55% of the weight to the average rating and the rest to the predicted compatibility rating, our model had an RMSE of ≈ 1.05 as opposed to the worse baseline model's ≈ 1.09.

Future Avenues

There are several avenues we can see to improve our project:

1. Include more information about the products (restaurants in this case) and reviews (e.g. the text of the review), in ways that don't affect the privacy of users. For example, review texts can be processed to remove any possibly identifying information before analyzing them.

2. Using a more complicated recommender system. The LFRS implemented is inherently linear, but in theory it is possible that ratings depend non-linearly on factors. Moreover, higher order terms would allow for interactions between factors.

3. Using a different KPI. For our goal, we care more about the recommended restaurants than the predicted rating, and so instead of RMSE on the ratings, we could measure the number of inversions between the actual ranked list of restaurants and the predicted ranking of restaurants produced by our models.