Yelp Based Recommendation Engine

Kushank Raghav

Problem Statement

"A recommendation engine is a feature (not a product) that filters items by predicting how a user might rate them. It solves the problem of connecting your existing users with the right items in your massive inventory (i.e. tens of thousands to millions) of products or content." [1]

My Goal was to create a recommendation engine using collaborative filtering approach.

More specifically, I aimed to do the following:

Predict the rating a user will give to a particular business and calculate difference between predicted ratings and observed ratings.

[1] http://www.datacommunitydc.org/blog/2013/05/recommendation-engines-why-you-shouldnt-build-one

DataSet

Yelp academic dataset was sourced from https://www.yelp.com/academic_dataset.

This dataset consists of various individual locations, such as restaurants, and Yelp users' reviews on products or services offered by various businesses using a one to five star rating system. The text reviews are also present in the database.

For collaborative filtering approach, a smaller dataset was used. This was done to reduce processing time. This dataset can be considered a representative sample of larger Yelp academic dataset. This dataset was sourced from a Kaggle Yelp business rating prediction challenge. It was sourced from https://www.kaggle.com/c/yelp-recsys-2013.

"This dataset is a detailed dump of Yelp reviews, businesses, users, and checkins for the Phoenix, AZ metropolitan area."[2]

The primary programming language that was used in the project is Python 2.7+.

[2] https://www.kaggle.com/c/yelp-recsys-2013/data

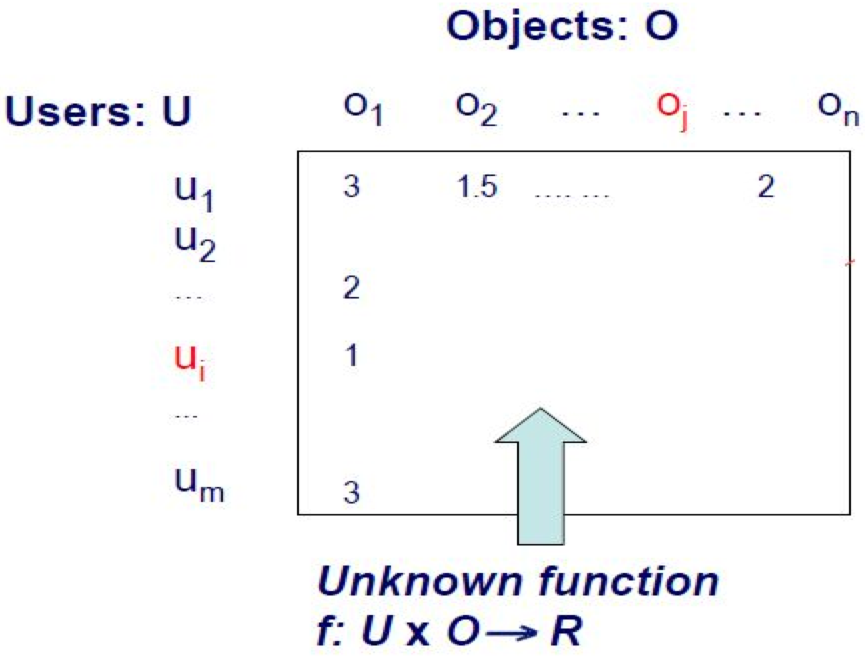
Methods

Collaborative Filtering approach

“Collaborative filtering algorithms take user ratings or other user behavior and make recommendations based on what users with similar behavior liked or purchased.” [3] Collaborative filtering predicts a given user’s preference based on the preferences of similar users. This approach works on two assumptions: Users who share a common interest will have similar preferences and vice-versa.[4]

One quantitative measure to encode a user’s preference is rating a user gives to an item. In the case of Yelp database, each user assigns a rating (1-5) to one or several businesses. These review objects are present in a JSON file. I manipulated this data to convert it to the following form:
{User1: { Business1: ‘rating’, Business2: ‘rating’....}, User2: { Business1: ‘rating’, Business2: ‘rating’....},....}

This format allows us to represent or view data in the following form:



[5]

[3] http://www.datacommunitydc.org/blog/2013/05/recommendation-engines-why-you-shouldnt-build-one
[4] SI 650 Week 8 Lecture
[5] SI 650 Week 8 Lecture

Similarity between any two users is computing using similarity measures. In our project, we used five similarity measures: “Pearson correlation similarity”, “Euclidean distance”, “Manhattan distance”, “Chebyshev distance”, and “Minkowski distance”. Pearson correlation coefficient was implemented in the code and the remaining distance measures were computed using in-built functions from scikit learn library. [6]

As an example, the given formula computes similarity between two measures:

The Pearson correlation similarity of two users x, y is defined as

$$\mathrm{simil}(x,y) = \frac{\sum\limits_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum\limits_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2 \sum\limits_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

where I_{xy} is the set of items rated by both user x and user y.
and $r_{u,i}$ is the rating user ‘u’ gives to item ‘i’ and $\bar{r_u}$ is the average rating of user u for all the items rated by u.[7]

It is important to note that distance measures should be converted to similarity measures by using the formula: $1/1+d$, where ‘d’ is distance. This ensures that similarity measure lies between 0 and 1. A ‘0’ similarity means that users are not at all similar and a ‘1’ similarity means that users are completely similar.

Prediction of a rating a user gives to an item is done using the following equation:[8]

$$r_{u,i} = k \sum_{u' \in U} \mathrm{simil}(u, u') r_{u',i}$$

$$k = 1 / \sum_{u' \in U} |\, \mathrm{simil}(u, u')|$$

Essentially, Value of rating a user gives to item ‘i’ is a weighted aggregation of similar users’ rating of the item. Weights correspond to similarity between the given user and the similar users. ‘k’ is the normalizing factor.

[6] http://scikit-learn.org/stable/
[7] http://en.wikipedia.org/wiki/Collaborative_filtering
[8] http://en.wikipedia.org/wiki/Collaborative_filtering

Evaluation

The data was divided into two sets: training and test. The split was 50%. I calculated ratings users gave to businesses using collaborative filtering approach. I compared these predicted ratings with actual ratings.

I wanted to know how much predicted ratings differ from actual ratings. Hence, I calculated root -mean-square error as an evaluation metric. RMSE is a “measure of the differences between values predicted by a model or an estimator and the values actually observed. Basically, the RMSD represents the sample standard deviation of the differences between predicted values and observed values.” [9]

Results are tabulated below:

Similarity Measures	**RMSE**
Pearson Correlation Coefficient	1.41992655709
Euclidean Distance	1.42117566441
Manhattan distance	1.42705461979
Chebyshev distance	1.42069228564
Minkowski distance	1.42117566441

[9] http://en.wikipedia.org/wiki/Root-mean-square_deviation