Find Your Food-mate

Abstract—We apply principles and techniques of recommendation systems to develop a predictive model of customers' restaurant ratings. Using Yelp's dataset, we have used content based features to identify customer and restaurant profiles.

I. Introduction

We think that people bond over food easily and in social networks are often unaware of people who share similar tastes, literally. From our own personal experience, we believe that people often bond over food, and so we want to recommend people of common food preferences and food-related activity. To do this, we aim to conduct an analysis of the Yelp network. To do so, we want to use Yelp user, business, and review data and find similar ratings between users. Our approach takes into consideration the Yelp user base, as well as user activity: creation of user reviews, ratings, cuisine and follows. Realistically speaking though, communities are much more likely to form when the members are in close physical proximity with each other, so our metric will also be sensitive to check-ins to the same restaurants, which gives us geographical data. The following proposal evaluates recommendation system. Advantages and disadvantages discussed range from the ability for an algorithm to capture belonging in multiple communities, to the computational viability of certain approaches for certain datasets.

- **Background**

We select some of popular restaurants of Chicago city from yelp whose followers belong to some ethnicity. We use this information to label the data and train a model. We call the popular restaurants as prime restaurants. First we came up with set of prime restaurants; we use these prime restaurants to label the data. Choosing these prime restaurants is very important, because these prime restaurants help us to learn other features of ethic group. The prime restaurants should be popular among particular ethnic group at-least 90% of followers who follows those restaurants should belong to same ethnicity and these prime restaurants should cover wide area set of audience. Of course it is difficult to come with the set of prime restaurants. With domain knowledge about that ethnic group it is possible. Once we came with prime restaurants, we sample 20 restaurants and 539 users of those restaurants

II. Data Collection

Beautiful Soup is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after Tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping. Beautiful Soup sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.

We have used Yelp to scrape data from http://www.yelp.com/chicago.The range of businesses is limited to locations in the area of Chicago, IL. Yelpers have written more than 90 million reviews by the end of Q3 2015. For our project we have limited ourselves to collect odd 600 reviews over a list of 40 restaurant .This can be extended to any number , by specifying the links of the webpages we are interested in collecting. We have a module which reads this .txt file and does the webscraping and save it as a CSV file .Apart from usernames and ratings , we have collected other data such as Cuisines of the restaurants.

III. Methods

We read the data locally stored from a CSV file each record. We make use of User-User Collaborative filtering . We have used sklearn.feature_extraction.text class to use various methods present in it. It converts a collection of documents to a matrix of token counts. This implementation produces a sparse representation of the counts using scipy.sparse.coo_matrix.

- fit_transform(raw_documents[, y]):To learn the vocabulary dictionary and return term-document matrix.
- get_feature_names(): It is used for Array mapping from feature integer indices to feature name.

We have used term **frequency–inverse document frequency**, which is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. This we have implemented in **document_frequencies** and **tfidf** methods**.**Once we have our TF-IDF ready we start creating user profiles based on their ratings in make_user_profiles method. Later we Compute the Euclidean norm of one row of a csr_matrix in **compute_norm** method. We compute the similarities in our users by using different measures such as **Cosine similarity , Jaccard similarity** and **Pearson Similarity**.Once we have these similarity scores we predict the unrated ratings of the users in **predict_ratings_w_user_profiles.Cosine similarity** is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

The **Jaccard distance**, which measures *dis*similarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or,

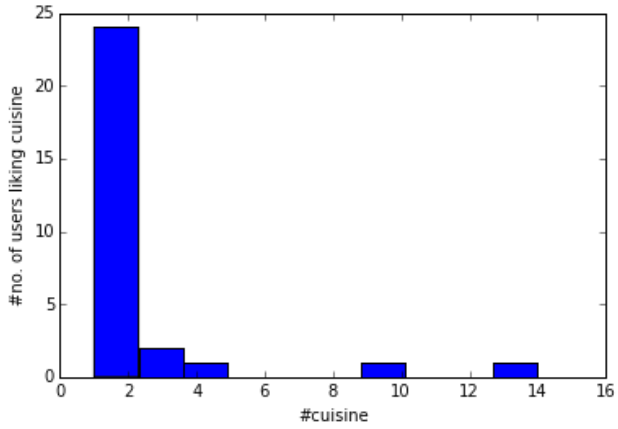
equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union:

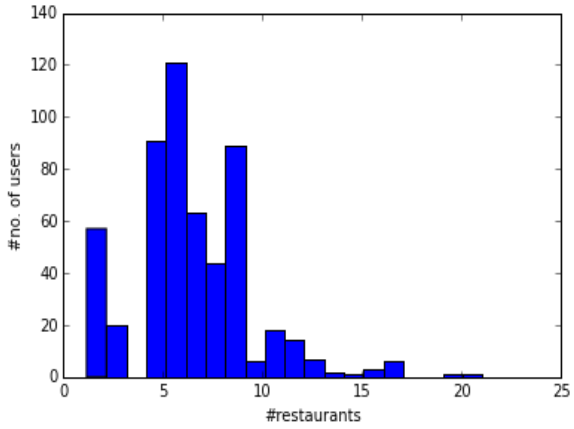
$$d_J(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}.$$

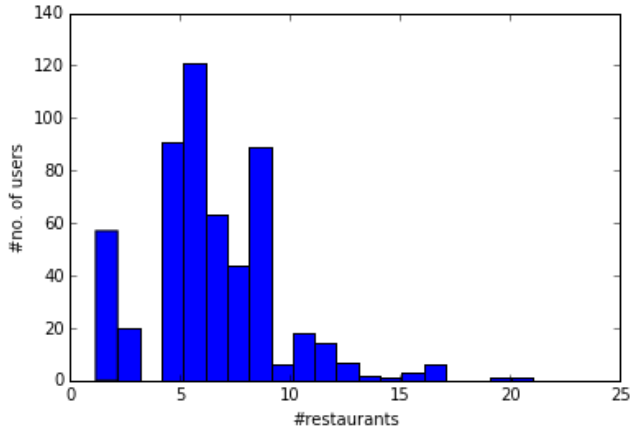
Collaborative filtering (CF) is a technique used by some recommender systems.[1] Collaborative filtering has two senses, a narrow one and a more general one.[2] In general, collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc.[2] Applications of collaborative filtering typically involve very large data sets. Collaborative filtering methods have been applied to many different kinds of data including: sensing and monitoring data, such as in mineral exploration, environmental sensing over large areas or multiple sensors; financial data, such as financial service institutions that integrate many financial sources; or in electronic commerce and web applications where the focus is on user data, etc.

IV. Experiments

We have experimented with our data and tried finding Cosine Similarities , Jaccard similarities and Pearson Correlation. Based on our experiment set we got better results when compared to Cosine similarity. We evaluate similarity scores based on which we can recommend friend recommendation. This works well when done on a large scale, as







V. Related work

In this paper Yelp recommendation system the approach used is Clustering and k-fold cross validation. Our approach is unique as we have used recommendation system. As it is in our curriculum and Collaborative filtering gives a proper scores and we feel it is a better approach.

VI. Conclusions and Future Work

Given the limited data set and lack of API usage , our data was restricted for smaller size and thus the model would perform much better as the size of the data increase. Pearson co efficient would give us a better answer. Latent Factor Models would be the best possible approach to go ahead with and lot of research is in progress.