Analysis of Yelp Dataset

Haoran Fang
Department of Electrical, Computer
& Energy Engineering
University of Colorado Boulder
Hafa8098@colorado.edu

Rui Xu
Department of Electrical, Computer
& Energy Engineering
University of Colorado Boulder
Ruxu5905@colorado.edu

Xinshuo Yang
Department of Applied Mathematics
University of Colorado Boulder
Xiya8696@colorado.edu

ABSTRACT

Yelp provides rating and review information about local restaurant, hotel, shopping and bar. The review contains star rating, short comment or pictures. Sometimes, these are not enough for people who want ratings and reviews on specific categories. So we decide to classify those regular reviews into different categories.

The Yelp dataset also contains the friend relationships between different users. With this big social network system, we would like to ask: what information can we mine from this graphical, interconnected data structure? Among the various graph patterns, the first thing we want to mine from this Yelp social network is communities -- which group of people are connected more than the other group of people.

Keywords

1. INTRODUCTION

Yelp provides rating and review information about local restaurant, hotel, shopping and bar. The review contains star rating, short comment or pictures. In this project, we consider the problem and application of mining information from the Yelp Dataset. This is currently an active and exciting area of research with a fairly rich depth. Overall, the problem can be stated in a simple way: given a set of local businesses (eg. restaurants, grocery stores, etc.) with some basic information (location, hours, parking, price, rate, etc.), and a set of users who write reviews and give rates on those local businesses, we would like to mine useful information from the dataset. In particular, the following questions will be mainly concerned in this project: community mining and review categorization. In the following sections, we will introduce the dataset and then describe the above two questions in detail.

2. YELP DATA SET

The data we will use in this project is from Yelp Dataset Challenge. Yelp provides this real-world dataset which includes information about local business, reviews and users from Phoenix, AZ, Las Vegas, NV, Madison, WI, Waterloo, CAN, Edinburgh, UK.

The followings are examples for business information, user information, and user's review, respectively:

{ "_id" : ObjectId("54c9cf888ecf62685da57f87"), "city" : "Edinburgh", "review_count" : 7, "name" : "Oriental Supermarket", "neighborhoods" : ["Tollcross", "Old Town"], "type" : "business", "business_id" : "BVxlrYWgmi-8TPGMe6CTpg", "full_address" : "125 Lauriston Pl\nTollcross\nEdinburgh EH3 9JN", "hours" : { }, "state" : "EDH", "longitude" : -3.2025293, "stars" : 3.5, "latitude" : 55.9441696, "attributes" : { "Accepts Credit Cards" : true, "Price Range" : 2, "Parking" : { "garage" : false, "street" : false, "validated" : false, "lot" : false, "valet" : false } }, "open" : true, "categories" : ["Food", "Ethnic Food", "Grocery", "Specialty Food"] }

{ "_id" : ObjectId("54c9d19a8ecf62685da7e904"), "yelping_since" : "2012-02", "votes" : { "funny" : 1, "useful" : 5, "cool" : 0 }, "user_id" : "qtrmBGNqCvupHMHL_bKFgQ", "name" : "Lee", "elite" : [ ], "type" : "user", "compliments" : { }, "fans" : 0, "average_stars" : 3.83, "review_count" : 6, "friends" : [ ] }

{ "_id" : ObjectId("54d053148ecf62734c479f65"), "votes" : { "funny" : 0, "useful" : 2, "cool" : 1 }, "user_id" : "Xqd0DzHaiyRqVH3WRG7hzg", "review_id" : "15SdjuK7DmYqUAj6rjGowg", "text" : "dr. goldberg offers everything i look for in a general practitioner. he's nice and easy to talk to without being patronizing; he's always on time in seeing his patients; he's affiliated with a top-notch hospital (nyu) which my parents have explained to me is very important in case something happens and you need surgery; and you can get referrals to see specialists without having to see him first. really, what more do you need? i'm sitting here trying to think of any complaints i have about him, but i'm really drawing a blank.", "business_id" : "vcNAWiLM4dR7D2nwwJ7nCA", "stars" : 5, "date" : "2007-05-17", "type" : "review" }

3. Generating Review Text Features

3.1 Small Dataset and Text Preprocessing

In order to extract the features from the review of each Yelp user, what we care about is how we can assign one topic for the word in the comment. Topic Modeling is an important tool we need to use during the whole project.

All the raw data is stored in a Json file. Since the data in original file is too huge, what we do first is extracting all the Chinese Restaurants from the corpus to test the correctness of the algorithm.

We need to select the features and construct the vocabulary so as to obtain the best performance. Here we filter all the review word by removing stopwords (which will be discussed below) and using stemming technology to reduce inflected (or sometimes derived) words to their word stem.

3.1 Keywords Selection

3.1.1 Eliminating StopWords

If a review is “I really like the service”. What we care about is the attitude of customer and the area referred. Here what we want is “like” and “service”, and we can filter the other words out. In computing, stop words are words which are filtered out before or after processing of natural language data (text). Normally, stopwords are some of those common, short, function words, such as *the*, *is*, *at*, *which*, and *on*. In this project, we use SKlearn Package in python to filter all the stopwords.

3.1.2 Stemming

Stemming is a common step for preprocessing. In order to reduce the size of the initial feature set is to remove misspelled or words with the same stem. A stemmer (an algorithm which performs stemming) removes words with the same stem and keeps the stem or the most common of them as feature. For example, the words “train”, “training”, “trainer” and “trains” can be replaced with “train”.

With Stemming, we can classify the words in each review with several roots so that the size of the review will be reduced.

The stemming procedure is implemented by the Snowball Algorithm. Snowball is a small string processing language designed for creating stemming algorithms for use in Information Retrieval.

Snowball is a small string-handling language, and its name was chosen as a tribute to SNOBOL, with which it shares the concept of string patterns delivering signals that are used to control the flow of the program. The basic data types handled by Snowball are strings of characters, signed integers, and boolean truth values, or more simply strings, integers and booleans. Snowball's characters are either 8-bit wide, or 16-bit, depending on the mode of use. In particular, both 8-bit ASCII and 16-bit Unicode are supported.

Here we show the example of how we select keywords from original user review.

> *“It's a chain but they have amazing service and delicious food. The portions seem to vary depending on how busy they are, but they're never tiny. Go with an appetite and a friend (or many) so you can share dishes!”*

This review is from the restaurant P.F Chang’s at Peoria, AZ. After performing the text preprocessing, we get the following data:

> *["chain", "amaz", "servic", "delici", "food", "portion", "seem", "vari", "depend", "busi", "theyr", "never", "tini", "go", "appetit", "friend", "mani", "share", "dish"]*

3.1.3 TF-IDF

Tf–idf, short for term frequency–inverse document frequency, 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[1].

For the Term Frequency, tf(f,d), the simplest definition is to choose the original frequency of a simple word. But in order to prevent bias towards longer documents, we have to choose the tf(f,d) in the following form:

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

Figure 1 [2]

For Inverse Document Frequency, it is a measure of how much information the word provides, whether this word is frequent or not.

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Figure 2 [2]

4. Topic Modeling

Topic models are a suite of algorithms that uncover the hidden thematic structure in document collections. These algorithms help us develop new ways to search, browse and summarize large archives of texts.

Topic models provide a simple way to analyze large volumes of unlabeled text. A "topic" consists of a cluster of words that frequently occur together. Using contextual clues, topic models can connect words with similar meanings and distinguish between uses of words with multiple meanings.

In order to find a latent subtopics in yelp reviews, we adopted online LDA, a generative probabilistic model for collections of discrete data such as text corpora. We present the breakdown of hidden topics over all reviews, predict stars per hidden topics discovered.

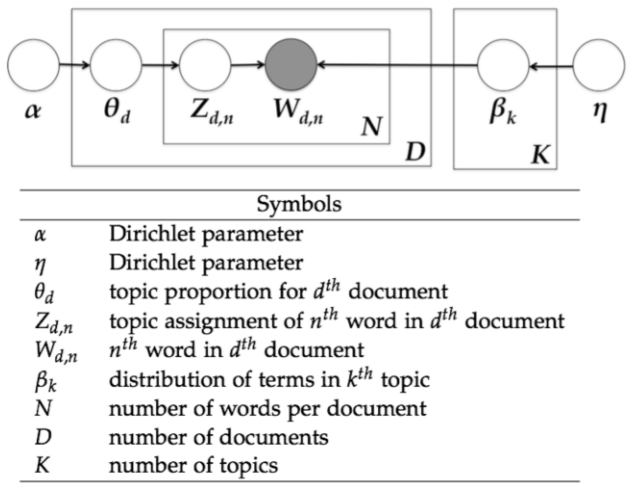
LDA can be expressed by plate notation, which defines the pattern of conditional dependence between the random variables. Latent random variables are depicted by unshaded circles, and observed random variables are depicted by shaded circles. Edges represent dependences between variables, and the rectangular plates represents replications[3].

Topic(‰)	food		service		food		food		environment	
	good	12	order	21	good	20	chines	15	dish	17
	soup	12	service	12	tri(tried)	13	food	15	order	14
	dish	11	beef	12	chines	11	dumpl	15	nice	10
	chicken	10	came	10	like	11	noodl	13	menu	9
	flavor	10	time	10	dumpl	11	good	12	sauc	8
	food	10	us	10	mama	11	order	11	one	8
	one	10	one	9	china	10	soup	10	im	8
	cold	9	check	9	soup	10	beef	10	fish	7
	place	8	roll	8	order	8	restaur	8	chines	7
	noodl	8	go	8	bun	8	fri	8	get	7

Table 1

Topic(‰)	food		service		food		food		environment	
	good	12	order	21	good	20	chines	15	dish	17
	soup	12	service	12	tri(tried)	13	food	15	order	14
	dish	11	beef	12	chines	11	dumpl	15	nice	10
	chicken	10	came	10	like	11	noodl	13	menu	9
	flavor	10	time	10	dumpl	11	good	12	sauc	8
	food	10	us	10	mama	11	order	11	one	8
	one	10	one	9	china	10	soup	10	im	8
	cold	9	check	9	soup	10	beef	10	fish	7
	place	8	roll	8	order	8	restaur	8	chines	7
	noodl	8	go	8	bun	8	fri	8	get	7
final stars	3.5		3.2		3.69		2.317		3.66	

Table 2



Symbols	
α	Dirichlet parameter
η	Dirichlet parameter
θ_d	topic proportion for d^{th} document
$Z_{d,n}$	topic assignment of n^{th} word in d^{th} document
$W_{d,n}$	n^{th} word in d^{th} document
β_k	distribution of terms in k^{th} topic
N	number of words per document
D	number of documents
K	number of topics

Figure 3 [3]

Parameters θ_d and β_k can updated using a variety of methods. Here, the LDA algorithm I utilize relies on collapsed Gibbs sampling, an inference technique that outputs $Z_{d,n}$, θ_d and β_k.

However, traditional LDA does not model the influence that ratings have on β_k. If reviews justify ratings and ratings generate the reviews, and thus the topics, then a more appropriate LDA would model the conditional dependence between a rating r and β_k. In other words, term distributions of a topic would be affected by the value of the star ratings.

Table 1 shows the top possible topics with top 10 words generated by LDA model. Topics and words are ranked by probabilities. Each topic has been manually labeled by interpreting the theme represented by the top words. We use the online word cloud generator to get a better interpretation which is displayed in Table 2.[4]

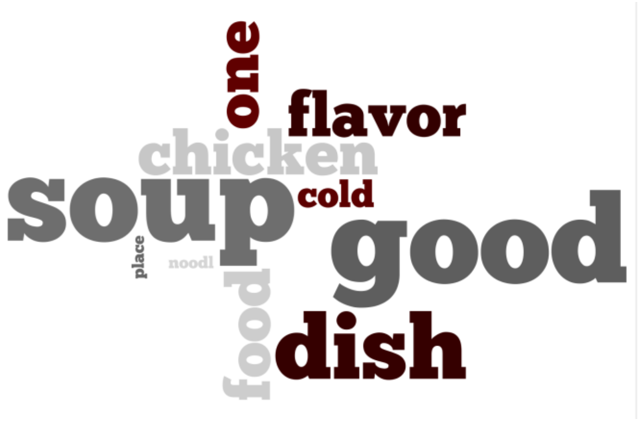


Figure 4

5. Sentiment Analysis

Think about two sentences:

I really like the food here, but the parking lot always crowded.

The parking lot always crowded, but I really like the food here.

You see, both sentence contains the same word, but we feel a little bit different since the order of the positive and negative emotional word differs. They already contain the same etyma after stemming, but we need a tool to analyze this precisely. Here, we use Sentiment Analysis.

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. It aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation, affective state, or the intended emotional communication.

When predicting the star rating per hidden topic, we need to figure out an approach to accurately predict sentiment for each topic in each review. For example, we have a 5-star review and we know the topics assigned to this review. The problem is we cannot tell if the topic is positive or negative. The 5-star review may state that "I love the food here, but I don't like the service". However, we haven't found an efficient way to generate two or more sentiments on one document. Instead, we use one sentiment analysis algorithm to generate a total score for each document. We use an python library which is called "*TextBlob"* developed by Steven Loria. "*TextBlob"* is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.[5]

The polarity score generated by sentiment analysis ranges from -1 to 1. The higher the score, the more positive it is. For example, I have a review looking like this:

"Unbelievably terrible company!! If there was a way to give negative stars I would! I cannot stress enough to NOT rent from this company or use them as a PM. It is a horrible company!"

The analysis will generate a polarity score of "-0.59375". Another positive review example will be like this:

"Oh my gosh, I have been here once so I don't know what I got. But I do remember I got the macaroni salad and it was the best. I crave it every time I pass by haha. I can't wait to go back again. I definitely recommend."

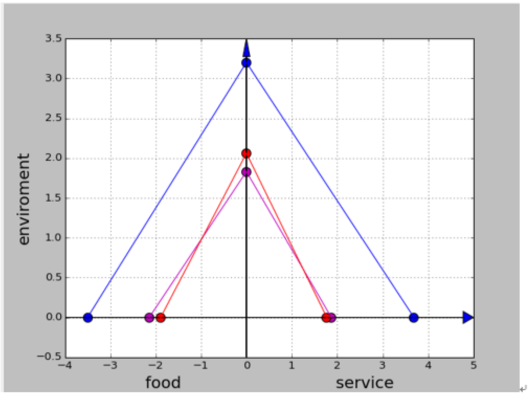
This review has a positive score of "0.3".

After the sentiment analysis, we search for all the user reviews related with a specific topic. We use the sentiment score as a weight to calculate the average stars for one specific topic.

The final score for each review in restaurant would be:

$$\frac{(SentimentScore\ \text{´}\ ReviewStar) - Min}{Max - Min}\ \text{´}\ 5$$

Figure 5 shows the final score for each topic in Figure 3.



🟦 China Mama Restaurant

🟪 Ling & Louie's Asian Bar and Grill

🟥 SUSHISAMBA Las Vegas

Figure 5

China Mama Restaurant: 3.5 stars overall.

Ling& Louie's Asian Bar and Grill: 3.5 overall.

SUSHISAMBA Las Vegas: 4.0 overall.

6. Discussion

In this project, our goal has been approached basically, although the accuracy cannot be guaranteed. What we have touched is the area that many scientists have reached for a long time. Natural Language Processing, Topic Modeling, Sentiment Analysis, which are the hottest topic these years., seem to be still have a large space to be improved.

We consider that if the technology we use is mature enough, then Yelp will come up a better idea to make their star system more perfect. We still have a long way to go because the language is so fantastic, we struggle in the situation in English. But how we can handle the situation in Chinese, Japanese and other more complicated language. How we can keep more feature after removing the stopwords and finishing the stemming. And we still have a lot to do.

Then we run our algorithm to all the Chinese Restaurants we have, for each subtopic, we draw a histogram , which are shown as follow:

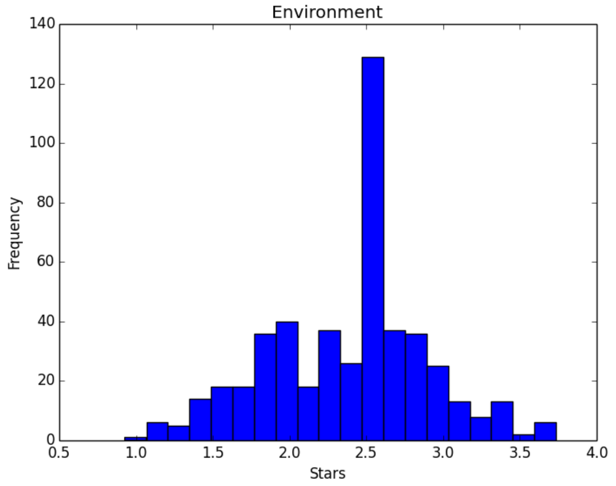


Figure 5: Environment

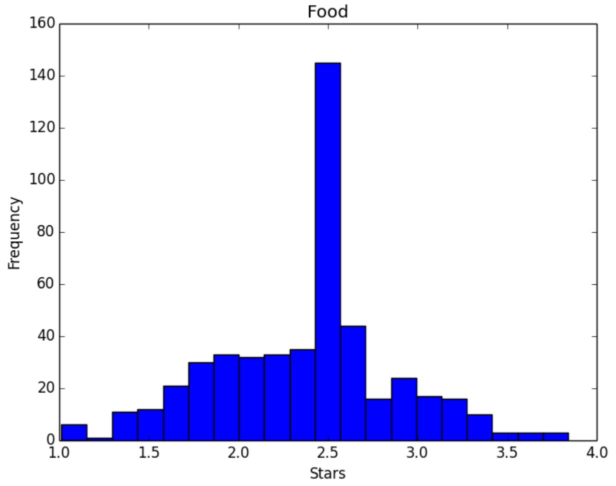


Figure 6:Food

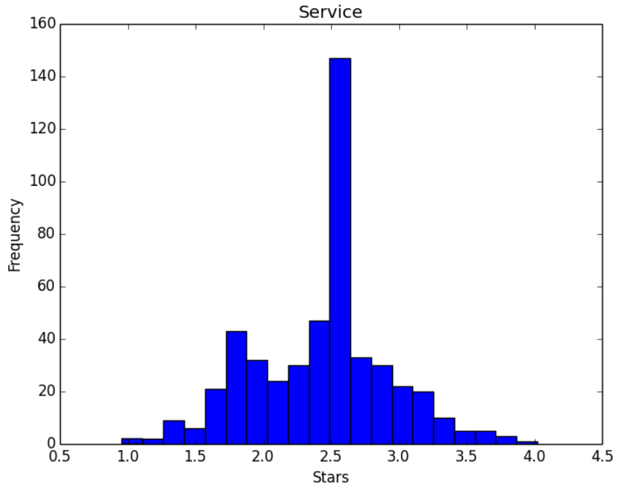


Figure 7: Service

Surprisingly, we find all these are nearly Gaussian Distribution, which is centered at 2.5. Which shows the reason that why 2.5 seems to be the most frequent result according to the test we did.

7. COMMUNITY MINING

7.1 Community Detection

The Yelp dataset contains 252898 users' information, and 123368 of them have at least one friend. Since this is a very large and complex social network, we randomly take 2000 of those who have at least one friend, and build the graph corresponding to it.

To analyze this graph, we use igraph tool with Python interface. It is a collection of network analysis toolbox, in particular, it has many community detection functions. We apply the following 3 methods on our graph: fastgreedy, leading_eigenvector, and multilevel.

Users on Yelp are usually friends with only a small subset of other users, which make this graph relatively weakly connected. Thus, most communities detected from any of the above 3 methods have very few members, in most case only one member in one community. However, we are more interested in those who have more members, because shared common properties among members only make sense in relatively large communities. For this purpose, we show the graphs with communities' sizes greater than 40 in Figure 5-7.

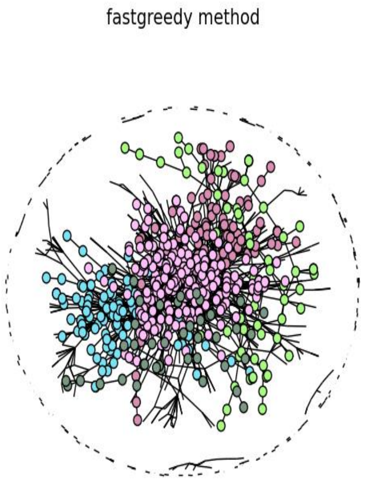


Figure 8: Fastgreedy Method

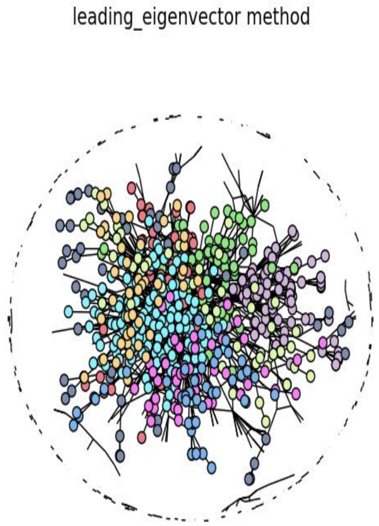


Figure 9: Leading_eigenvector Method

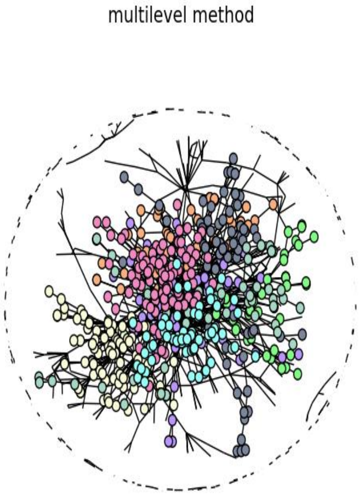


Figure 10: Multilevel Method

To evaluate the performance of a method, it essentially means applying it to a specific problem whose solution is known and comparing such solution with that delivered by the method. However, the real solution to our problem is not available and we don't have a direct way to test each of the methods we used. But we could compare two different methods by computing communities' distance measure. In this project, we use the variation of information of Melia(2003) as the distance measure. Table 3 gives the distance between different methods and it motivates us to use Multilevel method, because Multilevel has the least average distance.

Table 3: Comparison Between Different Methods

	Fastgreedy	Leading_eigen	Multilevel
Fastgreedy	0.0	0.387	0.282
Leading_eigen	0.387	0.0	0.367
Multilevel	0.282	0.367	0.0

7.2 Observe Local Behavior From Global Data

People on Yelp write reviews on local business (eg. restaurants). We can think about this as a way of grouping people. For those who write reviews on the same local business, they belong to the same group. We would like to see how communities we detected from friendship correlate with this. Table 4 shows, for communities with more than 10 members, how many of them have written reviews on the same local business, i.e. how many of them have ever been to the same local business at least once. We also give the total review number and ratio between them. From this we can see that for any of the communities we find from previous section, almost more than 10% of the members have written reviews on at least one local business.

Table 4: Review Number From Global Data

Local Business Number	Total Review Number	Number of Reviews From One Community	Ratio
1	66	4	6.1%
2	61	8	13.1%
3	68	10	14.7%
4	46	5	10.9%
5	79	11	13.9%
6	100	11	11.0%
7	42	5	11.9%
8	99	13	13.1%
Average	70.125	8.375	11.9%

What we have done so far is: (1) randomly picking 2000 users with at least one friend; (2) detecting communities among these users(divide them into groups); (3) find how many users in each community(group) would like to go(write reviews) to each local business. However, this might be very biased because we use the global data and try to see the local users' behaviors.

7.3 Observe Local Behavior From Only Local Data

To mine local users' behaviors from the local data, we take all the restaurants from Arizona. Then we take all the users who have ever written reviews to any of there restaurants. Finally we do the same procedure on these local data as what we have already done on the global data. The results are shown in Table 5.

Table 5: Review Number From Local Data

Local Business Number	Total Review Number	Number of Reviews From One Community	Ratio
1	16	2	12.5%
2	41	6	14.6%
3	23	3	13.0%
4	14	2	14.3%
5	52	8	15.4%
6	25	5	20.0%
7	29	5	17.2%
8	30	5	16.7%
9	29	7	24.1%
10	19	2	10.5%
11	15	3	20.0%
Average	26.63	4.36	16.2%

From table 5 we can see that, for any of the communities we detect, 16.2% of the members on average have written reviews on at least one restaurant. We think 16.2% is a relatively high percentage, compared with the large data size. So our results show that people who are friends with each other would like to go to the same restaurant.

8. Data Analysis

Given the community detection on the yelp social network, another question we would like to ask is that what other common behaviors do the users in each community have. For example, do people in the same community write more reviews than others, do people in the same community give higher rating than others? To analyze this, we use boxplot on the data of each community. Figure 8 shows the number of reviews for each community. The dotted line shows the average number of reviews each user write for the whole dataset. We can see that average numbers of reviews per user in each community are quite different. For example, users in community 7 write much more reviews (250 per user) than other communities, while users in community 8 write very few reviews (20 per user).

For users' ratings, we first user z-score normalization to map all the ratings to be normal distributed. We do this because yelp uses 0 (lowest) to 5 (highest) stars to measure how good a restaurant is, which is somehow hard to tell a customer like or dislike one restaurant if he/she gives it 3 stars. Then we use boxplot (Figure 9) again to show how average ratings in each community differ from others. We see community 2 and 11 are more likely to give positive reviews. However, overall the average ratings given in different communities are close. We cannot say that communities behave differently on giving ratings for restaurants.

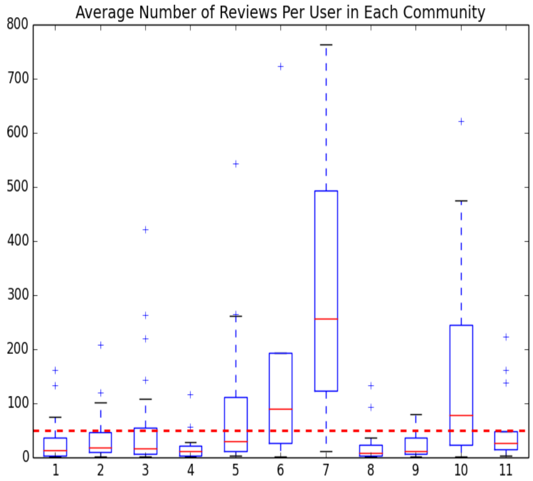


Figure 11: Average Number of Reviews Per User in Each Community

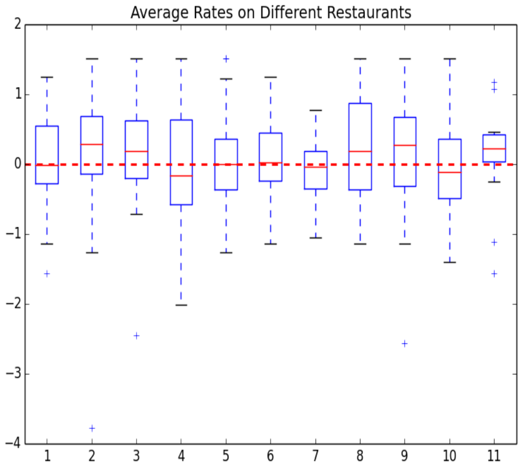


Figure 12: Average Ratings in Each Community

9. Conclusion

In this proposal, we first briefly introduce the Yelp dataset we will use in our project. Then we propose two main problems we will solve regarding to the dataset. Finally, we expect that using the approaches we described, we could categorize reviews into different categories, and find some interesting information by discriminating different communities.

For the first problem, from table 1 we can see that most of the reviews are talking about the food. Most of the top words are related with food. We also find lots of food related words in other restaurant's reviews which is natural because people care more about what they will eat in restaurants.

From figure 5, we can see that the subtopics of a 4-star restaurant may lower than 4. For the "SUSHISAMBA Las Vegas" example, the food, service and environment are all lower than 4 stars. This means users may give high stars on their reviews, but they may have a more neutral review.

So if we compare those three restaurants, we may choose the "China Mama Restaurant" because of its higher subtopic stars. We now compare the "Ling& Louie's Asian Bar and Grill" and the "SUSHISAMBA Las Vegas". "Ling" has a slightly higher environment score and "SUSHISAMBA" has a higher food score. Now we can choose which restaurant to go based on which subtopics we prefer.

10. REFERENCES

[1] Rajaraman, A.; Ullman, J. D. (2011). "Data Mining". Mining of Massive Datasets.

[2] Wiki pedia tf-idf.

[3] Linshi, Jack. "Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach."

[4] http://www.wordle.net/ word cloud generator.

[5] TextBlob Website: http://textblob.readthedocs.org/en/dev/.

11. Appendix

Work distribution:

Xinshuo Yang mainly works on the community detection part.

Haoran Fang mainly works on Topic Modelling, Sentiment Analysis, Preprocessing part.

Rui Xu mainly works on the data analysis, figure construction, statistics part.

All of us work for the proposal editing and the presentation slides.

*On my honor, as a
University of Colorado
at Boulder student,
I have neither
given nor received
unauthorized assistance
on this work.*