Using Yelp Reviews for Predicting Users' Ratings (Status Report)

Haoran Zhang
Computer Science Department
haz64@pitt.edu

Keren Ye
Computer Science Department
key36@pitt.edu

Abstract

In this project we want to predict Yelps star rating by applying machine learning techniques such as topic modeling, sentiment analysis, and supervised machine learning. While there are many works that predict the star ratings, our project focus more on utilizing the textual information.

1. Introduction

There are several dataset provided by the industrial companies for the purpose of research. In this project, we shall mainly focus on the Yelp dataset. The Yelp dataset challenge provides academics a good way for practicing machine learning algorithms. It provides not only the rating data, but also the users reviews. It makes it possible to leverage the abundant text information from users review. However, predicting the ratings by using the tranditional recsys technologies does not apply the informative textual corpus. Also, simple model such as bag-of-words makes strict assumption of the conditional independant, meanwhile, it suffers the problem of symnonyms which means different words may indicate the same topic. Moreover, it is hard to incorporate all of the features into a comprehensive framework.

In our work, we are planning to project both the users and the local bussiness into the topics feature space. There are two advantages: 1) comparing with the bag-of-words representation, the topics representation is more compact, which means it will not suffer the problem of data sparsity; 2) the topic model is easy for human to understand, thus in the future we can apply some semi-supervised labeling to label each of the topic classes. Then, we will schedule to incorporate this representation with extra features to predict the star rating of a new local bussiness given a specific user.

The contribution of our works includes: 1) we combine the topic modeling with some man-made strategies for handling the sentiments; 2) we apply a comprehensive and concrete framework for non-linear regression/classification problem.

2. Methodology

Our pipeline of processing the Yelp dataset includes the preprocessing step of the Yelp data, preprocessing of the Yelp review corpus, pos/neg topic modeling, supervised machine learning, and evaluation.

2.1. Preprocess the Yelp Data

The Yelp dataset contains 77,445 records of business, 552,339 records of unique users, and 2,225,213 records of reviews. For the purpose of applying our approach, we firstly preprocess the Yelp dataset.

For each of the distinct users in the users' records, we search for all of the reviews given by that user, merge these reviews with the user's basic attributes. Then store them into a local database. Because our approach rely highly on the textual information, i.e. the reviews given by the user, we get rid of the users' records that have less than 10 reviews. We gathered 40,233 records after filtering out inactive users.

For the business records, the textual information is abundant. Thus we have to select among tons of reviews towards the same business. Considering the inbalanced number among negtive and postive reviews, our strategy is to maintain the top 40 reviews (sorted by votes) both for the negtive (stars < 3) and positive (stars ≥ 3) sentiments of each distinct business.

2.2. Preprocess the Yelp Review Corpus

Tokenization Tokenize a document in to its atomic elements, so that we can perform more operation on the documents. For example, 'He likes to eat brocolli.' will become to ['he', 'likes', 'to', 'eat', 'brocolli', '.']

Stop words We need to cut off stop words from the corpus, such as conjunctions ('or', 'of', 'for') or the word 'the' are meaningless to a topic model. Thus, our tokenized corpus will change to ['likes', 'eat', 'brocolli']

Stemming To reduce similar words in our corpus, we perform stemming algorithm on the corpus. For example, 'runner' and 'running' have the similar meaning, but in different forms. Stemming could help us to reduce these terms to 'run'. Thus, the final corpus will looks like ['like', 'eat', 'brocolli']. 'likes' change to 'like' in our example.

2.3. Positive/Negative Topics Representation

In our approach, we are considering of using the topics distribution among positive sentiments and that among negtive sentiments to represent the attributes of the user and the business. In order to generate the representations, we have done some works described bellow.

Corpus of positive/negative sentiments I.e., extract positive reviews, and negative reviews. So far, we consider that, if user rate a business a 5 stars, the review is a positive review. In contrast, if the user rating is 1 star, the review is considered a negative review. Now, we have 2 different corpus, one is positive reviews corpus, and another is negative reviews corpus.

Bag-of-words representation To generate topic model, we need to use the frequency of each word occur in a documents, that is bag-of-words features. Thus, we need to generate the dictionary for the documents, and calculate the frequency of each word occur. The final input of topic modeling algorithm are looks like [(0, 1), (1, 0), (2, 1), (3,1), (4, 0)...]. This is a sparse matrix, so we may expect there are too many 0 occurance for one document.

Train the topic model We use Latent Dirichlet Allocation algorithm to help us generate two different topic models based on positive corpus, and negative corpus. Here are few examples of the result from different models.

Topics from positive topic model

```
Topic 1: 0.023*us + 0.021*order
    + 0.019*food + 0.019*wait + 0.018*tabl
    + 0.018*minut + 0.013*ask + 0.013*drink
    + 0.011*t + 0.011*time
Topic 2: 0.022*us + 0.015*manag
    + 0.015*ask + 0.008*rude + 0.007*staff
    + 0.007*told + 0.006*experi
    + 0.006*employe + 0.006*one
    + 0.006*friend
```

Topics from negative topic model

```
Topic 1: 0.023*us + 0.021*order + 0.019*food
    + 0.019*wait + 0.018*tabl + 0.018*minut
    + 0.013*ask + 0.013*drink + 0.011*t
    + 0.011*time
Topic 2: 0.022*us + 0.015*manag + 0.015*ask
    + 0.008*rude + 0.007*staff + 0.007*told
    + 0.006*experi + 0.006*employe
    + 0.006*one + 0.006*friend
```

We still can easily infer the topic from these result, the first topic is complaint the waiting time, and the second topic is complaint the bad service.

The reason why some words is not readable, is because the result of stemming step. It will cut off one or more characters from the words.

Inference of the positive/negative topic distribution If we apply our model on the new corpus, we can have the inference of the positive topics and negative topics. For example, we could have the following positive model output, [(0, 0.0123), (3, 0.3210), (4, 0.0603), (5, 0.0801), (8, 0.0308), (12, 0.0201), (13, 0.0566), (14, 0.2590), (15, 0.0190), (17, 0.0214), (18, 0.0984)] This means in this users reviews, the weight of positive topic 0 is 0.01, the weight of positive topic 3 is 0.32, and so on. Of course, we have the similar output for negative topic distribution.

2.4. Supervised Learning

Now, our problem becomes a supervised learning problem in the form of $\mathbf{X} \to Y$. However, it is not that clear how we can apply supervised learning techniques. At now, we have two databases, which are the Users info database and the Business info database. The attributes of the database can be presented as the following tables.

User's attributes	Brief
review_count	review count given by the user
average_stars	average stars the user given
votes_useful	votes by others
friends	number of friends that the user have
+ topics dist	topic distribution for the '+' reviews
- topics dist	topic distribution for the '-' reviews

Table 1. User's info database.

User's attributes	Brief
stars	average stars the user given
review_count	review count given by the user
categories	0/1 codings indicates the categories
+ topics dist	topic distribution for the '+' reviews
- topics dist	topic distribution for the '-' reviews

Table 2. Business' info database.

We shall generate the final feature representation from the users and the business database. However, generating the final feature representation is method related. We suppose to do experiments using Linear Regression, SVR, Factorization Machines and Neural Network. For the previous two approaches, we should construct $\phi(\mathbf{x})$ for getting rid of the non-linear features. For example, the $\mathbf{x}$ may be the concatenation of the Users representation and the Business representation, in this case, one choice of $\phi(\mathbf{x})$ could be [..., $cosv_{pos}, cosv_{neg}$, ...], in which $cosv_{pos}$ is the cos value between the positive topics vector of the user and that of the business, and $cosv_{neg}$ similarly. In contrast to the Linear Regression and SVR, the later two approaches can handle such kind of non-linear features, thus we just concatenate the original users and business features to form feature representation.

2.5. Evaluation

Our predicting data contains tons of tuples of (user_info, business_info, ratings). We seperate the dataset into training data and test data. Then, we train our model on the training data and evaluate it on the test data. We do 10-folds cross-validation on our dataset and using the average MSE, RMSE, and R-squared to evaluate our model.

The baseline algorithms we have chosen is to use the average ratings of the business or the average ratings of the user to predict. The results we get from the baseline is described in Table 3.

Algorithms	MSE	RMSE	R-squared
Avg ratings by user	1.1875	1.0897	0.2045
Avg ratings by business	1.2788	1.1308	0.1434

Table 3. The performance of the baseline algorithms.

3. Current Progress and Future Works

Tasks	TODO	DOING	DONE
Preprocess the Yelp Data			○
Preprocess the Yelp Review Corpus			○
Pos/Neg Topics Representation			○
Supervised Learning	○		
Evaluation		○	

Table 4. Current progress and future works.