CSE 190, Assignment 2: Report

UC San Diego, Spring Quarter 2015

Submitted 2015621

Ganesh Datta
gdatta@eng.ucsd.edu

Nazia Rizvi
nrizvi2@gmail.com

Martin Bjeldbak Madsen
mbma11@student.aau.dk

Abstract

In this report, we describe and sum up our experiments on attempting to predict categories of restaurants using the, as of this writing, current Yelp Dataset Challenge dataset.

We experimented with various feature representations of our data and models while trying to devise the most accurate prediction model for prediction a list of categories given a review of a restaurant.

Our findings show that our predictive model is excellent as a supplemental predictive tool that could be used to improve the experience of Yelp users and aid in the discovery of restaurants. Details of this will be further explained in this paper.

1. Introduction

In the past, we've seen various attempts to predict ratings given a review, and other such predictive tasks. However, from a practical standpoint, this is quite useless to Yelp since when a review is provided, a rating field is mandatory. However, we noticed in our dataset of choice that many Restaurants had a single category — 'Restaurants', or very few descriptive categories at all. With this observation in mind, we decided to devise a model that could predict more accurate categories of a restaurant accurately from metadata related to a review. We hope that this model could be used to add further categories to restaurants to increase visibility of eateries and intelligently improve the user experience when searching for specific types of restaurants.

We first begin by exploring the dataset and describing some of the choices we made with regard to data selection and filtering. This exploration gives us a starting ground to select a predictive analysis task.

The section proceeding this is the predictive analysis task we have selected, which is the prediction of categories of a restaurant given a review and restaurant metadata (restaurant location, reviewer friends, etc.). We describe in detail the models we experimented with when finding the most accurate predictor. Additionally, we delve into detail with regard to the features we decided to use in our predictive task, and how we represented these features. Then, we analyze the importance of these features and features we found to provide very little data to our models.

Finally, we compare our predictive task to a bag-of-words-based model described in [1].

1.1. Related Work

A similar study to our predictive task is the estimation of Points of Interest (POIs) with contextual information. A POI is a specific location such as a restaurant doctor's office, or a shop. Su Jeong Choi, Seong-Bae Park and Kweon-Yang Kim, propose a model to predict the type of POIs using two contexts, an internal and an external one [1]. The internal context is just the name of the POI, and the external context consist of documents that describe the POIs, typically user reviews. The external context fills in for what the internal context misses. The example presented by the Choi's et. al. is a furniture store called Flipp, which has an ambiguous name when it comes to predicting the type of business. From a part of a toy review, "This compact space is full of cool furniture and interesting design pieces, and Flipp hits that style perfectly", we can recognize that Flipp is a furniture store. The POIs are represented as a bag–of–words feature vector, and the predictor is tested on the Yelp dataset, which is the same one we are studying in this report. The model uses a support vector machine (SVM). According to their results, their proposed method

achieves 70.35% of accuracy for 914 POIs. Our task differs from this task in that only text features were used and that we used a stochastic gradient descent algorithm.

We did utilize their concept of using review text for the feature creation, however we did not use business name as that did not correlate with our predictive task.

2. Dataset

We chose the datasets from the current (2015) Yelp Dataset Challenge due to the fact that it has detailed features across many different activities on Yelp in many different cities. It is split into business, review, user, check-in, and tip datasets.

2.1. Attributes

The datasets consist of features from 1.6 million reviews, 61.000 businesses, 500 thousand tips, and 366 thousand users [2]. For our prediction task, we use the business, review, and user datasets. Each of their features are as follows.

Business Data

```
'type': 'business',
'business_id': (encrypted business id),
'name': (business name),
'neighborhoods': [(hood names)],
'full_address': (localized address),
'city': (city),
'state': (state),
'latitude': latitude,
'longitude': longitude,
'stars': (star rating),
'review_count': review count,
'categories': [(category names)],
'open': True / False,
'hours': {
  (day_of_week): {
    'open': (HH:MM),
    'close': (HH:MM)
  },
  ...
},
'attributes': {
  (attribute_name): (attribute_value),
  ...
}
```

Reviews

```
'type': 'review',
'business_id': (business identifier),
'user_id': (author user identifier),
'stars': (star rating, integer 1-5),
'text': (review text),
'date': (date, formatted '2011-04-19'),
'votes': {
  'useful': (count of useful votes),
  'funny': (count of funny votes),
  'cool': (count of cool votes)
}
```

Users

```
'type': 'user',
'user_id': (encrypted user id),
'name': (first name),
'review_count': (review count),
'average_stars': (like 4.31),
'votes': {(vote type): (count)},
'friends': [(friend user_ids)],
'elite': [(years_elite)],
'yelping_since': (date),
'compliments': {
    (compliment_type): (num),
    ...
},
'fans': (num_fans),
```

There are many intuitions and theories that can be tested out on this dataset. Below, we describe our intuitions behind the model.

2.2. Review Distribution

While exploring the datasets, we found that the vast majority of reviews consisted of restaurant reviews. This can be seen in the fig. 1, graphing business category distribution to the number of businesses.

With this in mind, we decided to focus on only restaurant reviews. We believed that we would be able to build a better predictive model since restaurants would (in theory) have many features in common, thus limiting the scope of our model.

Then, we looked at the restaurant category distribution (see fig. 2), to see whether there were outliers in the data.

These businesses were decently distributed, so no further filtering of the data was needed.

After filtering out all non-restaurant businesses & their reviews, we ended up with a dataset consisting of 21,892 restaurants and 990,627 reviews. The most popular category of restaurant was Burgers.

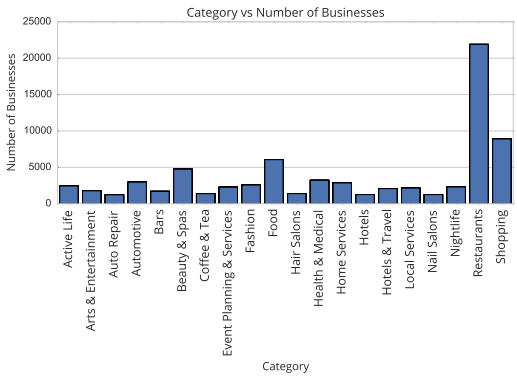

Figure 1: *Distribution of categories to number of businesses*

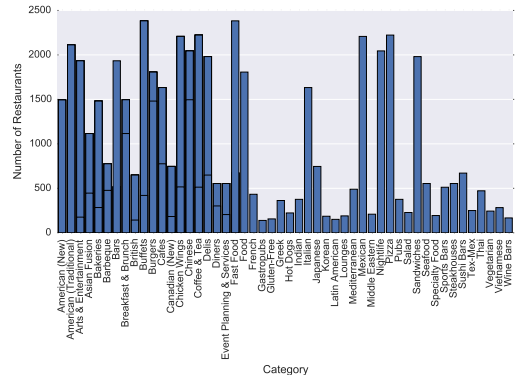

Figure 2: *Distribution of restaurant categories to number of restaurants*

However, the most reviewed category of restaurant was Nightlife, followed closely by Bars.

We then shuffled this data and split it into two datasets: 80% train and 20% test.

One interesting part of our dataset is that there was a small set of reviews not in English. We assumed that our text representation of the review dataset would find the important features of these languages as well — however, we would not be able to use any contextual features of the dataset due to this fact.

2.3. Chosen features

By selecting only restaurants and corresponding reviews, we were provided with some flexibility in terms of feature selection. We were able to choose features that we knew would work best for restaurant categorization.

The most obvious feature would be the actual review text. Due the the large number of words found in the overall corpus, we came up with a unique way of choosing a vocabulary. We split up the reviews in terms of category, and chose the top 500 unigrams and bigrams from each category. We added all these words to a set, and came up with a total vocabulary of approximately 50,000 words. We then found the tf-idf values for these words for each review.

An additional feature for the reviews was the price range of the business. While exploring the dataset, we found some correlation between the price and certain categories. As such, we decided to use the price range as one of the features.

Latitude and Longitude seemed to be a good set of features to use (scaled to a range of 0-1 with lat/90 and long/180). However, it turns out that the accuracy of our model increased for the most part with this addition to the feature but some predictions were rather questionable. Accuracy and the effects of these features will be described later in this paper.

Finally, we used the user information and their list of friends to construct an additional feature. Each user in the dataset has a list of friends. Our thought process was that a user is likely to review/eat at types of restaurants their friends reviewed. So, we go through their friends and count how many friends have reviewed a business of each category.

3. Model

Our extremely large training set (around 800,000 reviews and 350,000 users) provided us with many challenges. It is obvious from our problem that we need some sort of classification model, and that too a multi-class classification model since a restaurant can belong to various categories. Our first thought was to utilize an SVM as a classifier, except we found that it is not designed for multi-class tasks and it does not scale well to extremely large datasets such as the Yelp dataset.

We then found that Stochastic Gradient Descent (SGD) scaled extremely well to both large datasets and large number of features. At any given time, our feature vector for every review consisted of close to 50,000 entries. However, we still needed to figure out a multiclass classifier. We decided to use a one-vs.-rest classifier [**?**], which in essence creates one base classifier per label, and trains each of these different classifiers. Then, when running predictions, it combines the predictions of the various classifiers and combines them into a single multi-class prediction. We used a SGD classifier with squared hinge loss as the base classifier for our one-vs.-rest classifier.

We then designed the following score for our predictions.

$$
\begin{aligned}
f &= \text{Ratio of predictions with 100\% correctness} \\
b &= \text{Ratio of blank predictions} \\
r &= \text{Average correctness ratio} \\
\text{Score} &= 0.8f - 0.5b + r
\end{aligned}
$$

Here, we give the highest weight to predictions that are 100% correct, which is where the categories predicted are exactly the same as in the label. Then, we want to penalize situations where no categories were predicted. Finally, we normalize the score with the average corrrectness ratio ($\frac{\text{num correct categories}}{\text{num categories in label}}$).

Due to the nature of the dataset, we designed a baseline where we go through all the categories found in the training set. If the category, as a string, exists in the review text, we add that category to the list of predicted categories. The reasoning behind this baseline is that reviews tended to have relevant words — a review for a burger joint contains the word 'burger', for example.

We had multiple hyperparameters to tune with regard to the SGD Classifier. We landed on an α value of 0.0005, and chose squared hinge as the loss function being optimized.

3.1. Results

We found that many of the correlations noticed in the data exploration turned out to positively impact our predictions. By using only a **tf-idf** feature matrix for our predictions, we got a score of -0.9 due to the many predictions that were blank (approximately 49,000 blank predictions). However, the number of correct predictions confirmed our suspicions that the text of the review would be where the meat of the data would be.

Then, we used the price range of the business. Although our exploratory analysis of the dataset found some correlation between price range and category, using price range as an additional feature turned out to have predict more blank predictions, and the number of fully correct predictions dropped by a few thousand. However, some of our predictions were "parent" categories, such as predicting "Bar" for something that was actually a "Gastropub" — close, but not exactly correct. However, this is one of the more interesting things we noticed in our analysis of the model.

Latitude/Longitude seemed the most intuitive in terms of category prediction. One would expect certain categories to be more prominent in given geographical areas. This intuition turned out to be correct. Our score jumped to 0.15, with 30,000 fewer blank predictions. However, although the majority of categories predicted would be correct, some of the predictions were very bizarre, and turned out to be highly correlated with the latitude/longitude data.

Increasing the number of popular n-grams chosen from 500 to 1000 also increased the accuracy of our model.

4. Conclusion

4.1. Optimizations

Our model was highly complex, using around 800,000 reviews for training, and the training matrix consisted of over 50,000 columns. This resulted in many challenges when optimizing the model. At peak usage, our model consumed around 16GB of RAM while calculating.

One of the first optimizations we did was to move from an SVM based classifier to an SGD based classifier due to the higher scalability of an SGD classifier. Additionally, we utilized sparse matrices to save memory since we noticed that the tfidf matrix consisted of largely 0's, and only 100 values

for each review. This saved enormous amounts of memory. Additionally, parallelizing the one-vs.-rest classifier by using 8 threads further improved the performance.

One of the hardest problems to solve was the problem of scaling our features. Latitude ranges from $-90 \rightarrow +90$, and longitude from $-180 \rightarrow +180$. This provided quite an issue for scaling. However, since the tf-idf was calculated using a logarithmic representation, that feature set did not provide as much of a challenge in terms of scaling.

To optimize our tf-idf calculation, we used the top 500 n-grams from each category (as explained in the feature section of this paper). However, given more time, instead of choosing the top 500 words, we could have calculated the tf-idf for all words and chosen the most important words with this information rather than simply choosing the most popular ones. Increasing to 1000 improved accuracy but slowed down the process greatly. This could also be overfitting, by choosing far too many extra words that simply undermined the model instead of helping it.

4.2. What didn't work

We found that looking a user's friends did not improve the accuracy of our predictor in any way. This, we presume, is due to the fact that users are not interested in their friends tastes, and we did not take into account the ratings given by certain users to categories.

Additionally, we did experiment with using an SGDRegressor instead of a SGD Classifier due to the fact that each sub-classifier in the one-vs.-rest classifier was performing a regression task. Surprisingly, the classifier worked with much more accuracy. Time constraints did not allow us to further experiment with the regression model.

4.3. Interesting Findings

Through the analysis of our model, we came across a highly interesting piece of information — in many cases, our predictive model was far more accurate than the labels provided in the Yelp dataset. For example, Yelp did not categorize many restaurants with a bar as a "Bar". However, our predictor was able to correctly predict this category (confirmed by cross verifying with the Yelp page for the business and reading reviews/pictures).

Additionally, it correctly added categories such as Nightlife, Breakfast and Brunch, etc. to many businesses, which we were able to cross verify on yelp.com. Although our model was not 'accurate' in terms of the labels provided in the dataset, we feel that the practical applications of this predictive model can be far more useful than simply predicting categories accurately to the label.

For example, Yelp could use this feature to predict additional possible categories for businesses, allowing users to further explore categories and improve search results and categorization of restaurants. By providing a list of possible "new" categories for a restaurant and asking users to confirm whether they believe they are accurate for the restaurant, Yelp can increase discoverability of restaurants and improve the user experience. Additionally, this predictive model can be used to add categories to the 300 or so business in the dataset that had no categories other than "Restaurant".

In essence, we found that our predictive tool could be used as a supplement — accuracy would then have to be crowdsourced rather than calculated by using a mathematical formula.

4.4. Future Work

Below we describe setbacks and things we could see possibly improving the model if we had more time.

Many words were misspelled. We attempted to use a spell checking library in the preprocessing stage to correct review texts, but it made the model less accurate (E.g. it corrected "steak and fries" to "steak and fires"). The model could have benefited from a more robust spell checker.

Should incorporate the supplied check-in dataset as an additional data source. This might help the social feature, because users are much more likely to simply check in rather than spend time writing an entire review.

Add more features that might help improve classifier. Bag-of-words model from review text used by [1], among others, seems to have helped classification accuracy.

Limiting the amount of categories even farther than we already have. Could only concentrate on the top x most popular categories, since categories can be sub-categories of each other (e.g. the category "gastropub" is intuitively a sub-category of "pub")

References

[1] S. J. Choi, S.-B. Park, and K. yang Kim. Estimating category of pois using contextual information. *Indian Journal of Science and Technology*, 8(S7), 2015.

[2] Yelp. Yelp dataset challenge, 2015. `http://www.yelp.com/dataset_challenge/`.