Predicting Success of Restaurants in Las Vegas

Sang Goo Kang and Viet Vo
Stanford University
sanggookang@stanford.edu
vtvo@stanford.edu

Abstract— **Yelp has played a crucial role in influencing business success as it provides public information on the overall quality of businesses to customers. Using the Yelp open dataset from the Yelp Dataset Challenge, we extracted restaurant attributes and unigrams and bigrams from reviews to use as features for classification and regression to predict the star rating of restaurants in Las Vegas. The algorithms used for prediction were linear regression, SVR, SVM, and perceptron neural networks. Analysis on the test set shows that neural networks and SVM performed the best with classification accuracies of 48% and 42% respectively, which are about 4 times better than random guessing as we are dealing with a 9-class classification problem. We found that textual features which includes bigrams and unigrams were best in achieving low classification error during prediction compared to restaurant attributes.**

I. INTRODUCTION

The well-being of many businesses today heavily rely on the positive ratings given by their customers. With the founding of Yelp in 2004, the relationship between businesses and their customers has become more dynamic. Many businesses for example, offer special deals for visitors using Yelp, and previous visitors offer valuable advice for future customers based on their experience such as recommendations and warnings on what to purchase. In this project, we will utilize the Yelp public dataset to analyze the success of restaurants in Las Vegas. In particular, we will predict the star ratings of restaurants and find the most useful traits in determining their success. This task is important as it will allow new businesses with limited customer input to have a better idea of how well they will perform in the long run. This prediction will give restaurants the opportunity to improve their services at an earlier stage in their business. For our input features, we will use a restaurant's characteristics (hours open, food category, review count, etc.) and n-grams extracted from customer reviews. These features are inputted into our SVM and perceptron neural networks for classification, and linear regression and SVR for regression. The output for these algorithms is a star rating prediction of each restaurant.

II. RELATED WORK

There have been many works dedicated to analyzing the success of businesses based the Yelp dataset. One interesting method that has been used focuses on extracting subtopics from Yelp reviews and predicting a star rating for each subtopic [3]. Using an online Latent Dirichlet Allocation algorithm and Expectation Maximization, reviews were grouped into topics such as service, healthiness, lunch, etc., and a rating was assigned to each topic. This would allow a business to pinpoint its weaknesses by improving upon the topics that had the lowest ratings. This method however, suffers from the fact that it is difficult to verify the ratings assigned to each subtopic due to the unsupervised nature of the algorithm.

Another interesting study predicted business ratings by adopting a latent factor model for a business and its geographical neighbors [4]. It was shown that there was a weak positive correlation between a business's ratings and its neighbors' ratings. By incorporating geographical information, it was shown that the proposed methods had an improved rating prediction accuracy. This method was very effective, but can improved by including other environmental factors surrounding the businesses such as the ethnic community, traffic, etc.

One paper attempted to predict business star ratings based on business attributes such as noise level, smoking options, price range, etc. Linear regression, decision trees, and neural networks

were used to predict the business' rating [5]. It was discovered that the attributes were not very good for the predictive task. A better strategy would have been to utilize the user reviews in conjunction with the restaurant attributes.

Using Naive Bayes is a common technique used for text classification and had moderate success when performed on bigrams and unigrams extracted from user reviews [6]. Multi-class logistic regression is another algorithm that have been proposed to predict the star ratings of a review [7]. It performed well on unigrams, bigrams, and trigrams, but could be improved by using Parts-Of-Speech tagging, which we implemented in this project.

III. DATASET AND FEATURES

A. DATASET

The dataset used in this project came from the Yelp Dataset Challenge in the form of JSON files [1]. From this, the restaurants in Las Vegas were extracted, along with all the reviews associated with them. Given each restaurant, the initial data gave us the restaurant's characteristics as well as its reviews. The total amount of data for Las Vegas included 6764 restaurants. The data was split into 5764 training examples and 1000 test examples. The training examples were used for both cross-validation as well as model fitting, and the test examples were used to test the fully trained models.

B. PREPROCESSING

Initially, the characteristics were all preprocessed to have numeric values, such as times being changed from 11:30AM to 11.5, True/False being changed to 1/0, etc. In addition, any feature for which the restaurant had no information was given the value of -1. The feature "categories", which was a list of categories that the restaurant fits into, was treated as a sparse vector, with value 1 for each category that a given restaurant includes (Japanese, Food Truck, Seafood, etc.). Additionally features that appeared in less than 10% of the restaurants in the training data were filtered out to prevent excessive unnecessary features.

Restaurant reviews were preprocessed using the python *nlkt* natural language package. We first performed a Parts-Of-Speech tagging of all words in all the reviews, and filtered our all the words that were not adjectives. We then created a Bag of Words that contained a mapping from adjectives to the number of times they appear in our entire training dataset. This allowed us to extract the top k most used adjectives in all the reviews. These top k adjectives are then used as features for each restaurant. For each restaurant, we mapped each adjective to its frequency of appearance in the reviews of the restaurant. Finally, we normalized all the values. A similar approach was taken with bigrams, with the included constraint that each bigram must contain at least one adjective.

Fig. 1. Unigrams of adjectives from 5-star reviews

C. FEATURES

The final feature set included attributes of restaurants that were mostly not missing in the dataset, along with textual features like unigrams and bigrams. The unigrams were composed of the most common adjectives used in reviews, while the bigrams were the most common paired words in the reviews that contained at least one adjective. *Figure 1* shows an example of a word cloud that contains the top adjectives used in 5-star reviews. Words such as 'good,' 'delicious,' and 'favorite,' were found to be very popular among these reviews, which indicates that these may be very useful features for our supervised learning algorithms. The word cloud provided a high level overview of the usefulness of using popular adjectives as features. However, due to the fact that some adjectives can be paired with a negative word such as "not good", using the unigrams without any bigrams features do not represent the data as well as when both are used simultaneously.

IV. METHODS

A. LINEAR REGRESSION

Linear regression was used to attempt to fit the data without any additional feature processing. It was used to minimize the following cost function for which $h_\theta(x) = \theta^T x$, where $h_\theta(x)$ is used to generate new predictions.

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \tag{1}$$

B. SUPPORT VECTOR REGRESSION WITH RBF KERNEL

Support Vector Regression was also used to attempt to fit the data with a regression using a rbf kernel. The objective function for the SVR was gathered from previous work [2], in which $k(x_i, x_j)$ is the rbf kernel, described in *Equation 3*.

$$\arg\max_{\alpha,\alpha^\star} \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \alpha_i^\star)(\alpha_j - \alpha_j^\star)k(x^{(i)}, x^{(j)}) \\ -\epsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^\star) + \sum_{i=1}^{m} y^{(i)}(\alpha_i - \alpha_i^\star) \end{cases}$$
$$\text{subject to } \sum_{i=1}^{m}(\alpha_i - \alpha_i^\star) = 0 \text{ and } \forall i, \alpha_i, \alpha_i^\star \in [0, C] \tag{2}$$

$$k(x^{(i)}, x^{(j)}) = \exp\left(-\frac{||x^{(i)} - x^{(j)}||^2}{2\sigma^2}\right) \tag{3}$$

In order to make a new prediction, the trained model then follows equation 4.

$$f(x) = \sum_{i=1}^{m}(\alpha_i - \alpha_i^\star)k(x^{(i)}, x) + b \tag{4}$$

C. SUPPORT VECTOR MACHINE WITH RBF KERNEL

In addition to the regression models, classification models have also been used. In our case, the output number of stars of a restaurant can be classified in bins of 1, 1.5, ..., 5 stars. In this scenario, 9 SVM classifiers can be generated, such that each classifiers determines how likely an example is to be a specific class. The SVM risk function can be represented as follows, where K is the rbf kernel matrix for the data.

$$J_\lambda(\alpha) = \frac{1}{m}\sum_{i=1}^{m} L(K^{(i)^T}\alpha, y^{(i)}) + \frac{\lambda}{2}\alpha^T K\alpha \tag{5}$$

To make a classification, we use a one-vs-all method with the SVM models to generate predictors for each new example. *Equation 6* shows how kernelized SVM makes predictions on a new example x.

$$f(x) = \sum_{i=1}^{m}\alpha_i k(x^{(i)}, x) \tag{6}$$

D. PERCEPTION NEURAL NETWORK

The final classification model that was used is the perceptron neural network. A neural network attempts to model in a similar way to how the human brain solves problems. It is able to emulate this behavior through the use of interconnected nodes, in which each node represents a simple function such as a sigmoid or perceptron. An artifical neural network is shown in *Figure 2*. In this case, the green nodes are the input nodes, red are the hidden nodes, and blue are the output nodes. In a perceptron neural network, each of the hidden and output nodes represents a simple perceptron function, while each connection represents a linear multiplication by a weight in the weight matrix.

For simplification, the neural network was designed with 2 hidden layers, with the number of nodes being tuned as a hyperparameter. In this case, we have 3 weight matrices (with θ representing a weight) that are used as transformation between each layer. To generate classifications on a new example, Equation 7 is used, where $\sigma(x)_i = 1\{x_i > 0\}$ is the output of the perceptron.

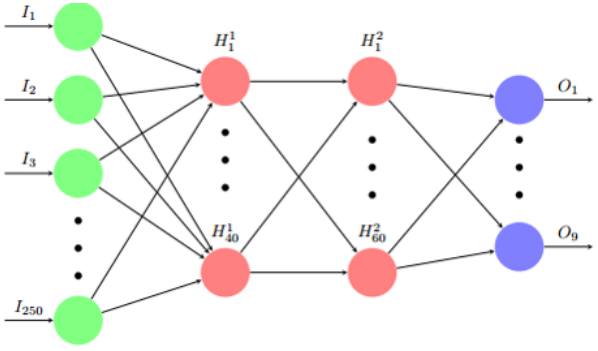

Fig. 2. Artificial Neural Network with two hidden layers

$$output = \arg\max_i \sigma(\theta_3^T\sigma(\theta_2^T(\sigma(\theta_1^T x))))_i \tag{7}$$

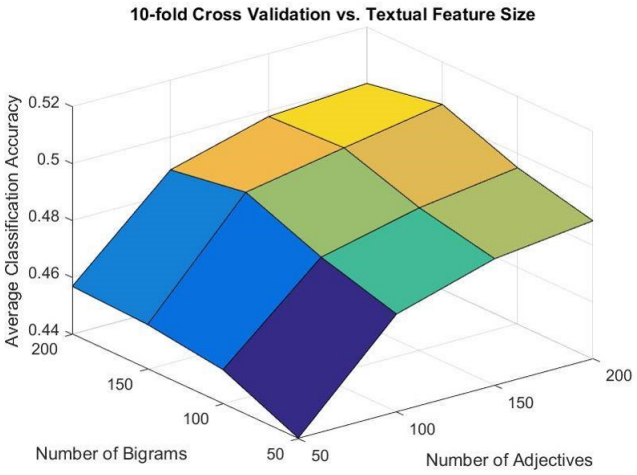

Fig. 3. Distribution on cross validation accuracy sweeping textual feature size

V. Experiments/Results/Discussion

A. FEATURE SELECTION

For the restaurant reviews, 10-fold cross validation was used to determine how many of the top adjectives and top adjective bigrams are to be used as review features. The results of the cross-validation can be seen in *Figure 3*. From this, we could set the feature set for the text reviews to be the 200 most common adjectives along with the 150 most common adjective bigrams.

B. HYPERPARAMETERS

Cross validation was also used in order to tune the hyperparameters of all the other algorithms. 10-fold cross validation was used in order to get an estimate on the test error just from using the training data. For the SVR model, cross validation was run on every combination of $\epsilon \in \{10^{-3}, 10^{-2}, 10^{-1}, 0.5\}, C \in \{10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5\}$ to find the most optimal combination. The same was done for the SVM, though it only swept through all the C values. For the neural network, the network was initialized with 50 nodes for both the first and second layer. With these parameters, the regularization parameter α was tested for $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5\}$. With the most optimal α value, the number of nodes in the first and second layer were swept in order to find the most optimal network configuration given the α. Every combination for the number of nodes in each layer where $numNodes \in \{20, 40, 60, 80, 100\}$ were tested. The results of the hyperparameter tuning can be seen in Table 1.

TABLE I
HYPERPARAMETERS

Restaurant Characteristics	
Hyperparameter	Value
C (SVR)	1
ϵ (SVM)	0.5
C (SVM)	1
α (NN)	0.0001
NN Layer 1 #nodes	20
NN Layer 2 #nodes	40

Review Text (Adjectives and Adjective Bigrams)	
Hyperparameter	Value
C (SVR)	1000
ϵ (SVM)	0.5
C (SVM)	10000
α (NN)	0.001
NN Layer 1 #nodes	40
NN Layer 2 #nodes	100

C. RESULTS

Each model was trained using the 5764 training examples, and the performance of each was tested using the test set of 1000 examples. Two error evaluation metrics were used. The RMS Error from *Equation 8* evaluates both how accurate and how close the example was to the true value. Classification error from *Equation 9* evaluates how accurate the model is with a zero-one loss function. For the regression models, classification error was computed by comparing the actual classification with the output rounded to the nearest 0.5. The results from each algorithm and both feature sets are shown in Table II. Due to the fact that there are 9 classifications, random guessing will result in a classification error of 0.89.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - f(x^{(i)}))^2} \tag{8}$$

$$\text{Classification Error} = \frac{1}{n}\sum_{i=1}^{n} 1\{y^{(i)} \neq f(x^{(i)})\} \tag{9}$$

TABLE II
RESULTS

Restaurant Characteristics				
Algorithm	Train Classification Error	Train RMS Error	Test Classification Error	Test RMS Error
Linear Regression	0.339927121	0.469367941	0.814	0.839642781
SVR	0.314766615	0.42980753	0.800	0.844393273
SVM	0.257331251	0.478498287	0.814	0.856446145
Neural Networks	0.307305223	0.473737672	0.768	0.849117189

Review Text (Adjectives and Adjective Bigrams)				
Algorithm	Train Classification Error	Train RMS Error	Test Classification Error	Test RMS Error
Linear Regression	0.251344786	0.31993053	0.594	0.563249501
SVR	0.247961131	0.332725456	0.597	0.538052042
SVM	0.174041298	0.281022463	0.583	0.562138773
Neural Networks	0.154606976	0.232256881	0.521	0.521296461

When the characteristics of the restaurant is used as the feature set, The Neural Network and SVR are shown to have the lowest test classification error (0.764 and 0.8, respectively). However, the Test RMS Error are minimized by the Linear Regression and SVR, with about 0.8396 and 0.8444, respectively. These results do indicate that these models perform better than random guessing, albeit not significantly so.

When the 200 most common adjectives and the 150 most common adjective bigrams in review text was used as a feature set, the algorithms were shown to perform significantly better than when simply using restaurant characteristics. The Neural Network and SVM models had the lowest test errors with 0.521 and 0.583, respectively. The Test RMS Error are minimized by the Neural Networks and SVR, with about 0.5213 and 0.5381, respectively. This feature set was shown to be significantly more representative of the number of stars of a restaurant as opposed to the characteristics of the restaurant.

VI. CONCLUSION/FUTURE WORK

From the results, we can note that the perceptron Neural Network is the highest performing algorithm for both feature extractors. This was because many features most likely would not have a simple relationship with the actual number of stars, and the neural network is able to generate complex relationships through the network of perceptrons.

Future work includes the use of unsupervised learning algorithms in conjunction to the supervised learning algorithms. This is because with the use of algorithms such as k-means clustering, the model is able to fit more closely to different geographic regions. Although a large k in this case would cause severe overfitting, a reasonable k value could result in a more accurate model due to differences in customer's desires depending on the region. We also can try other classification algorithms like naive bayes and random forests for text classification.

REFERENCES

[1] Public data: http://www.yelp.com/dataset_challenge

[2] Smola, Alex J., and Bernhard Schlkopf. "A Tutorial on Support Vector Regression." Statistics and Computing 14.3 (2004): 199-222. Web.

[3] Huang, James, Stephanie Rogers, and Eunkwang Joo. "Improving restaurants by extracting subtopics from yelp reviews." iConference 2014 (Social Media Expo) (2014).

[4] Hu, Longke, Aixin Sun, and Yong Liu. "Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction." Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM, 2014.

[5] Farhan, Wael. "Predicting Yelp Restaurant Reviews." UC San Diego, La Jolla (2014).

[6] Wang, Junyi. "Predicting Yelp Star Ratings Based on Text Analysis of User Reviews."

[7] Asghar Nabiha, "Yelp Dataset Challenge: Review Rating Prediction." 2016.