Task 6 Report

Changhun Park

Abstract

For processing the review text on a restaurant into features, I tried two types of text representing models: Term Frequency-Inverse Document Frequency(TF-IDF) and Topic Modeling using Latent Dirichlet Allocation(LDA). In addition to the review texts, I used for features the additional data such as zip code, review count, average rating and the cuisines provided by the restaurant. For classification of restaurants, I tried two types of learning algorithms: Logistic Regression and Support Vector Machine.

1. **Description and Comparison of Methods**

1.1 Text Representation Methods

For both TF-IDF and LDA, I represented the review texts in the file "hygiene.dat".

- TF-IDF: I tokenized the review texts on all the restaurants after removing stop words, and constructed them into a TF-IDF matrix with dimension 13299 by 96175. Thus, the review text on a single restaurant was represented as a TF-IDF vector with the length of 96,175.

- Topic Modeling by LDA: I also tokenized the review texts on all the restaurants after removing stop words and stemming. I used LDA for topic modeling algorithm with the number of topics as 10. Thus, the review text on a single restaurant was represented as a topic vector with the length of 10.

1.2 Feature Representation of Additional Data

After processing the review texts, I combined each of them with the additional data I got from the file "hygiene.dat.additional".

I first constructed an array from the raw data in the file of which row contains the list of provided cuisines, zip code, number of reviews, and the rating (0 to 5) for each restaurant.

Then, since the zip code and the cuisine list have categorical values, I made one-hot encoding for the two to mark the presence or absence of the category with 1s and 0s. The length of one-hot encoding for the zip code is 30, and that for the cuisine list is 98.

Finally, I normalized the number of reviews and the average rating to have values between 0 and 1 since the two have values greater than 1.

Below is the final structure of the feature table composed of TF-IDF and additional data:

TF-IDF vector with length 96175	One-hot vector for cuisine list with length 98	One-hot vector for zip code with length 30	Normalized Number of reviews	Normalized Average Rating

Below is the final structure of the feature table composed of topic vector and additional data:

Topic vector with length 10	One-hot vector for cuisine list with length 98	One-hot vector for zip code with length 30	Normalized Number of reviews	Normalized Average Rating

As a result, I got a total of 96305 columns of the feature table for TF-IDF and 140 columns for LDA. The feature table is divided into the two: X_train for training and X_test for evaluating. X_train is made out of the first 546 rows and X_test out of the rest 12753 rows. y_train is made out of the first 546 rows in the file "hygiene.dat.labels".

1.3 Learning Algorithms

I tried Logistic Regression and SVM algorithms for classification. The X_test with TF-IDF had to be divided into several files and evaluated one by one due to the memory problem since it is too big to process all together. The results of the several X_test files were combined into one for submission to the leaderboard.

Logistic regression: I tried the model, *LogisticRegression* from *sklearn.linear_model* with default parameter values and maximum number of iterations as 200.

SVM: I tried the SVM model using *SVC* class from *sklearn.svm* with kernel set up as 'linear'.

1.4 Comparison

A total of four F1 scores are shown in the table below for two different text representation methods of TF-IDF and LDA, and two different learning algorithms of Logistic Regression and SVM. The F1 score is the highest for the text representation with LDA and the learning algorithm with SVM.

	Logistic Regression	**SVM**
LDA	0.7377	0.7541
TF-IDF	0.345	0.6041

When comparing the text representation methods, LDA achieved higher F1 scores than TF-IDF. This is because the TF-IDF vectors can be very sparse, especially for a large vocabulary. In this case, the size was 96,175. If the vocabulary size is big, each word or token in the TF-IDF vector can little contribute to the output. In addition, it can be inferred that each term frequency is not much related with the hygiene of a restaurant. On the other hand, LDA, by focusing on a smaller number of topics in this case 10, produces a denser text representation, resulting in much contribute to the output. In addition, the topics with which reviewers are concerned can be much highly related with the hygiene of a restaurant.

When comparing the learning algorithms, SVM achieved higher F1 scores than Logistic Regression. This is because SVM is more suitable in binary classification. SVM creates decision boundaries based on support vectors, which are the most informative data points. Logistic Regression, on the other hand, uses all data points to fit a probabilistic model. This difference can lead SVM to create more vigorous decision boundaries in the situations where there are noisy data points. In addition, as Logistic Regression outputs a continuous value between 0 and 1, we classify the output as binary classes, which can result in inaccuracy.

2. Details on the Best Performing Method

The best performing method was using LDA as the text representation method and SVM as the classification learning algorithm.

2.1 What toolkit was used?

All toolkits are from the python packages. For both learning algorithms, the toolkit *StandardScalar* was used when standardizing the features.

LDA: *LdaModel* from *genism.models*

TF-IDF: *TfidfVectorizer* from *sklearn.feature_extraction.text*

Logistic Regression: *LogisticRegression* from *sklearn.linear_model*

SVM: *SVC* from *sklearn.svm*

2.2 How was text preprocessed?

During the text preprocessing, I tokenized the review texts with the toolkit *word_tokenize* from *nltk.tokenize*. I also used *STOPWORDS* from *gensim.parsing.preprocessing* to filter out common sop words such as 'the', 'is', 'you', 'they', 'in', '$', '#', ';', '&', '.', 'i', '..', '...' ','('.')'. Additionally, I stemmed the tokens using the toolkit *PorterStemmer* from *nltk.stem* to decrease the number of tokens when modeling topics using LDA.

2.3 How was text represented as features?

I first considered using sentiment analysis to represent review texts, but it took too much time for the Stanford Core NLP server to process all the sentences in the review texts for sentiment analysis. Therefore, I chose LDA and TF-IDF as text representation methods. For LDA, I represented the review text for a single restaurant as a topic vector with size of 10. In other words, the review text was embedded in a dense vector. For TF-IDF, I represented the review text for a single restaurant as a tf-idf vector with size of 96175. In other words, the review text was embedded in a large vector. However, the result is that the size of embedding is not important, but the content of embedding is important.

2.4 What was the learning algorithm used?

I considered many learning algorithms for classification models such as Logistic Regression, SVM, Random Forest, and Neural Network. However, due to the time constraint, I used SVM and Logistic Regression from sklearn package in python to get the F1 score from each learning algorithm. The result showed that SVM is good for binary classification problem.