SMAI

Restaurant Recommendation System

Introduction

There are many recommendation systems available for problems like shopping, online video entertainment, games etc. Restaurants & Dining is one area where there is a big opportunity to recommend dining options to users based on their preferences as well as historical data. Yelp is a very good source of such data with not only restaurant reviews, but also user-level information on their preferred restaurants. In yelp data we will restrict to restaurants segment within the business category as recommendation is a very good fit in that system. In this project we aim to build a model that recommends restaurants to users. We explore the use of different machine learning techniques and also features that perform well on this classification.

Problem

The way the problem is modeled is to predict yes/no for any given restaurant and user. Using Yelp's dataset, we extract collaborative and content based features to identify customer and restaurant profiles. To recommend a restaurant to a user we use different machine learning techniques and also features that perform well on this classification.

One way this model could be used in practice is by having an automatic 'Recommend: Yes/No' message when a user visits a restaurant's profile page.

Dataset

The data that we used in this project was obtained from the Yelp Dataset challenge. The dataset contains five different tables: User, Business, Review, Check-In and Tips. The data has 27257 restaurants, 552339 users, 55569 check-Ins, 591864 tips and 2225213 reviews.

From this Yelp dataset, we took the latest 1 month data as test dataset. Apart from the test dataset, last 3 months data was taken as training dataset. Remaining part of the data apart from the test and training is used for calculating derived features(historical data).

Features

Given that our input tuple is <user, restaurant> I have features of following categories:
a) User-level features
b) Restaurant-level features
c) User-Restaurant features

The features that we developed to solve this problem are mentioned below .The highlighted features in the below datasets are selected for training the data.

Raw Features

Business Data Features(bold are the selected features)

‘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, rounded to half-stars),
'review_count': review count,
'categories': [(localized category names)]

'open': True / False (corresponds to closed, not business hours),
'hours': {
(day_of_week): {
'open': (HH:MM),
'close': (HH:MM)
},
},
'attributes': {
(attribute_name): (attribute_value),
},
}

User Data Features(bold are the selected features):

{
'type': 'user',
'user_id': (encrypted user id),
'name': (first name),
'review_count': (review count),
'average_stars': (floating point average, like 4.31),
'votes': {(vote type): (count)},
'friends': [(friend user_ids)],
'elite': [(years_elite)],
'yelping_since': (date, formatted like '2012-03'),
'compliments': {
(compliment_type): (num_compliments_of_this_type),
...
},
'fans': (num_fans),
}

Review Data Features(bold are the selected features):

{
'type': 'review',
'business_id': (encrypted business id),
'user_id': (encrypted user id),
'stars': (star rating, rounded to half-stars),
'text': (review text),
'date': (date, formatted like '2012-03-14'),
'votes': {(vote type): (count)},
}

Derived Data Features:

1) User-level: Average historical rating from this user, # of reviews
2) Business-level: Average historical rating for this business, # of reviews
3) Average rating of that user on that category given the current restaurant's category(Fast food, Buffets) .
4) Average rating of that user on that attribute given the current restaurant's attribute (Parking garage, Caters).

Pre-Processing

- The latest year reviews are separated from the the whole review dataset from which the last 1 month data is held out as the test data and 3 months data is taken to be the training data . The remaining data is considered as historical data for calculating the derived features.
- For every review in the review data , a label has been assigned . A label can either be "Yes" or "No" . This is calculated by taking two parameters into consideration. One is the rating star given by the user to that restaurant and other is the review text written by the user.

For every review , a score which is a sum positive and negative scores is calculated by using wordnet senti analysis . Wordnet give a score for every word in the review and a positive,negative score is generated for each review.

If the rating is greater than 4 , then a “Yes” label is assigned irrespective of the wordnet score . If the net wordnet score is negative and the rating is less than 3 , then a “No” label is assigned . If the net wordnet score is positive and the rating is greater than 3 , a “Yes” label is assigned.

- Duplicates in the business and user datasets are removed.
- Every business in the business dataset have some binary attributes like parking,alcohol,desserts,etc .There are a total of 61 attributes that are taken as features.

These attributes vary from business to business , so to assign a value for every attribute that is missing in a particular business data , “Collaborative Filtering “ procedure is used.

Collaborative filtering, also referred to as social filtering, filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. A person who wants to see a movie for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see.

- All features that have been mentioned above in each dataset are separated . For a particular review , the features corresponding to that user are taken and also the features corresponding to that business are taken and joined . This is done for every review and final features input file with a label is created.

Classifiers

- **Linear SVM** : We trained and tested our dataset using Linear SVM classifier.

Linear SVM is the newest extremely fast machine learning (data mining) algorithm for solving multi class classification problems from ultra large data sets that implements an original proprietary version of a cutting plane algorithm for designing a linear support vector machine. Linear SVM is a linearly scalable routine meaning that it creates an SVM model in a CPU time which scales linearly with the size of the training data set.

- **SVM with RBF kernel :** We trained and tested our dataset using SVM with RBF kernel classifier.

RBF network can be used find a set weights for a curve fitting problem. The weights are in higher dimensional space than the original data. I Learning is equivalent to finding a surface in high dimensional space that provides the best fit to training data. I Hidden layers provide a set of functions that constitute an arbitrary basis for input patterns when they are expanded to the hidden space; these functions are called radial basis functions.

- **Logistic Regression :** We trained and tested our dataset using logistic regression.

Logistic Regression is a special type of regression where binary response variable is related to a set of explanatory variables, which can be discrete and/or continuous. The important point here to note is that in linear regression, the expected values of the response variable are modeled based on combination of values taken by the predictors. In logistic regression Probability or Odds of the response taking a particular value is modeled based on combination of values taken by the predictors.

- **Random Forest :** We trained and tested our dataset using random forest.

Random forests is a notion of the general technique of random decision forests[1][2] that are an ensemble learning method for classification,regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

- We also built a new classifier method using **Neural Networks** for calculating the weights for the features .
    - A net weight score is calculated for each of the available restaurants in the dataset.
    - Net Weight score for every restaurant is calculated by taking all the features of that user and restaurant that are mentioned above
    - Weights are given to each and every taken feature.
    - A weight for every feature is found by backpropagation of the training data (data of last few months) by building a neural network.
    - Given a user(U) and a restaurant(R), one should find out whether the user likes it or not.
    - All the restaurants for which that particular user has given reviews are considered and an average net weight score is calculated.
    - This calculated average net score is taken as a threshold(t).
    - The net weight score(w) for this restaurant(R) will be known as the weight scores for all the restaurants are calculated in the beginning.
    - If we find w to be greater than the taken threshold t , we return a ‘yes’

saying the user will like this restaurant.

- If we find w to be lesser than the taken threshold t , we return a ‘no’ saying the user will not like this restaurant.

Challenges

- The review dataset that have been downloaded does not have pre built labels so these should be generated by us.
- Labels are generated by using the rating and the wordnet score that is generated for the review which is a challenging task.
- Collaborative Filtering is a challenging task which is used for restaurant attributes as these vary from restaurant to restaurant . A binary value for the missing attributes is calculated for such attributes using collaborative filtering.
- Data Partition for training and testing the data is a difficult task considering the large data available.
- Derived Features are calculated by using the historical data . Average historical rating from a user based on the restaurant attributes and category preferences and for business have been calculated.

Results

Training Samples : **103610**

Test Samples : **23948**

LogisticRegression :

- Correct results : 18331
- Incorrect results : 5617

Accuracy: 76 %

Confusion Matrix :

13605	3887
1730	4726

Linear SVM :

- Correct results :14986
- Incorrect results : 8962
-

Accuracy: 62 %

Confusion Matrix :

6424	51
8911	8562

RandomForest:

Correct Results : 17640

Incorrect Results : 6308

Accuracy : 73 %

Confusion Matrix :

12365	3338

2970	5275

12365 3338
2970 5275

SVM with RBF Kernel:

Correct results : 16977

Incorrect results : 6971

Accuracy: 70%

Confusion Matrix :

14075	5711
1260	2902

References

- http://cs229.stanford.edu/proj2014/Ashish%20Gandhe,Restaurant%20Recommendation%20System.pdf
- **Dataset :** https://www.yelp.com/dataset_challenge