ORIE 4741 Project Proposal

September 22, 2016

Siyuan Wang Mingchen Zhang Xuan Zhao

Yelp provides crowd-sourced reviews about local businesses to users, which leads the right customers to the right businesses. The mode brings mutual benefits to both customers and business owners. However the current issue with Yelp reviews is that they cannot provide customized recommendations based on a person's taste for food and entertainment, therefore looking for the exact services we want might be time-consuming. In this project, with the learning algorithms we acquired from this course, we wish to establish a recommendation engine that could push up businesses to the user's preferences and habits. In this way the customers could find the local businesses they are looking for more efficiently, and the business owners could improve their strategies by learning more about the target customers.

In this project, we hope to construct separate models for the two objectives we have. The first model will aim at predicting how a potential user may rate a given business by using the background data sets of the business and users (Data frames in the Appendix). With the prediction results from the first model, the second model will focus on how to set up a hierarchy system of potential high rating users for businesses to build up their advertising strategies, and users could at the same time obtain reliable customized businesses recommendations. The data set we will utilize come from Yelp data set, which provide a large volume of features for both business and users.

In order to construct a prediction model for how users may review unknown businesses, we need to incorporates features related to various attributes of the business and the user behavior. With the "user" data as well as the corresponding "review" and "check-in" data, we can extract features such as the the neighborhood of the restaurants, categories of businesses the user favors in his review, and other potential properties of the restaurant that may make the user give positive review for the business. The breadth of the features and feature engineering techniques we can choose from, as well as the size of the test set we are allowed to have made us believe that our approach will likely to yield a successful model.

After we have proved that our prediction model has sufficient accuracy, we will then be able to generate review predictions of businesses that the user has not visited and use them to construct our recommendation engine. The accuracy of our prediction model will dictate the effectiveness of the recommendation system. As long as we can prove that our prediction model can produce accurate results, we will be able to gain enough confidence that our recommendation engine is providing promising business push-ups or target customers' information.

Appendix

1 Business

```
    {
        '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),
            ...
        },
    }
}
```

2 Review

```
{
    '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)},
}
```

3 User

```
{
    '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),
}
```

4 Check-in

```
{
    'type': 'checkin',
    'business_id': (encrypted business id),
    'checkin_info': {
        '0-0': (number of checkins from 00:00 to 01:00 on all Sundays),
        '1-0': (number of checkins from 01:00 to 02:00 on all Sundays),
        ...
        '14-4': (number of checkins from 14:00 to 15:00 on all Thursdays
            ),
        ...
        '23-6': (number of checkins from 23:00 to 00:00 on all Saturdays
            )
    }, # if there was no checkin for a hour-day block it will not be in
       the dict
}
```