Predicting Whether A Particular Customer Will Write A Review of A Particular Restaurant in A Particular Month using One-class SVM by Modeling Visiting Incentive And Review Incentive

Dataset

Raw datasets are JSON files from the *yelp dataset challenge*:
http://www.yelp.com/dataset_challenge

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),
        ...
    },
}
```

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)},
}
```

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),
}
```

Analysis

To build a prediction model, I chose SVM since it is one of the best off-the-shelf classifiers. In intuition, we also need to put customers' visiting incentive into consideration since a visit always precedes a review.

The model I plan to build will 'studies' a customer's behavior as to why he/she chose to visit a particular restaurant and had the incentive to leave a review on yelp's website. However, using binary-class SVM might not fit in this task, because what we can observe are restaurants, which a customer has visited and left reviews; we cannot presume that a custom does not want to visit or to leave a review of another restaurant just because we have not observed such instance. In light of this, I am proposing to use One-class SVM (OSVM) to 'study' the behavior of individual customers.

Modeling

I model a customer *u*'s action of reviewing (review_score) of a restaurant *i* as the sum of the customer's incentive to visit *i* and to leave a review:

$$review\_score = visting\_incentive + review\_incentive$$
$$= (p\_u \cdot q\_i) + (r\_u \cdot k\_u) = w.x$$

where '.' means dot product, *q_i* is the properties of restaurant *i* and *k_u* is the properties of user *u*:

$$q\_i = [i\_stars, review\_count\_i, latitude, longitude, food\_type, review\_month]$$
$$k\_u = [num\_friends, num\_fans, review\_count\_u, votes, elite?, average\_stars]$$
$$x = [q\_i, k\_u]$$

Label is positive if $w.x > b$ and negative otherwise. This is equivalent of an SVM setting.

The above vectors can be obtained from Yelp's JSON files. OSVM will learn the offset *b*, *p_u* (e.g. how a customer responses to *q_i*) and *k_u* (e.g. does a person tend to leave more reviews if he/she has more friends/fans?)

Primary I will investigate only *active customers*, namely, those who have left at least 100 reviews. Also, to ensure restaurants are mainly visited by local customers, I chose restaurants from Charlotte, NC. If we study restaurants from Las Vegas, NV, for example, chances are we will observe a lot of swing-by customers and be unable to study their behaviors.

Data Preprocessing

According to what mentioned above, I have constructed a data frame of size 7036 x 163, which consists of 42 *active customers* and the corresponding data from the Charlotte area. Categorical data are all binarized. Continuous values are standardized. In conclusion, this data frame is mixed of *continuous and boolean data*.