CS 412: Intro to Machine Learning
Progress Report

Branches

We have divided up our project into two branches to approach our problem from two different directions. The main task is to predict a user's rating of a business. The two approaches are based on different aspects of the data: one focuses on the text and date of a review, while the other focuses on the user's profile, including their previous ratings and friend base.

1. **Profile-Based Branch:**

   Hongwei Jin

   Krutarth Joshi

   Ashwin Sattiraju

   Zhan Shi

2. **Text-Based Branch:**

   Dan Zhao

   Natawut Monaikul

   Aayush Kataria

Accomplishments

Our main accomplishments so far have to do with cleaning the data and extracting features from the data.

1. **Profile-Based Data:**

   The initial data obtained from the Yelp Dataset Challenge website was in JSON format. We converted the data into CSV files to make it easy to work with in Python. We extracted about 90 features from the dataset. Out of 90 features, the top 20 most useful features will be used for developing our model.

   - Original Features:

Accepts Credit Cards, Accepts Insurance, Ages Allowed, Alcohol, Attire, BYOB, BYOB/Corkage, By Appointment Only, Caters, Coat Check, Corkage, Delivery, Dogs Allowed, Drive-Thru, Good For Dancing, Good For Groups, Good For Kids, Happy Hour, Has TV, Noise Level, Open 24 Hours, Order at Counter, Outdoor Seating, Price Range, Smoking, Take-out, Takes Reservations, Waiter Service, Wheelchair Accessible, Wi-Fi, attr_casual, attr_classy, attr_divey, attr_hipster, attr_intimate, attr_romantic, attr_touristy, attr_trendy, attr_upscale, average_stars, city, comp_cool, comp_cute, comp_funny, comp_hot, comp_list, comp_more, comp_note, comp_photos, comp_plain, comp_profile, comp_writer, elite, fans, friends, goodfor_breakfast, goodfor_brunch, goodfor_dessert, goodfor_dinner, goodfor_latenight, goodfor_lunch, latitude, longitude, music_background_music, music_dj, music_jukebox, music_karaoke, music_live, music_playlist, music_video, open, parking_garage, parking_lot, parking_street, parking_valet, parking_validated, pt_amex, pt_cash_only, pt_discover, pt_mastercard, pt_visa, res_dairy-free, res_gluten-free, res_halal, res_kosher, res_soy-free, res_vegan, res_vegetarian, review_count, stars, state, votes_cool, votes_funny, votes_useful, y_rate, yelping_since

- Most Related Features(by PCA, anova):

  By Appointment Only, Good For Groups, Noise Level, Open 24 Hours, Wi-Fi, attr_upscale, average_stars, goodfor_breakfast, goodfor_brunch, goodfor_dessert, goodfor_latenight, goodfor_lunch, parking_lot, pt_cash_only, res_vegetarian, review_count, stars, votes_cool, votes_useful, y_rate, yelping_since

2. **Text-Based Data:** The data was divided into a training set and test set. We plan to use cross-validation, but we would like to run a full iteration on one split before doing several. As each reviewer can also leave a "tip" for a business (a short, typically one-liner to sum up their thoughts) alongside their full review, we first matched up tips to reviews and added that text to the review text for feature extraction. Here are the features we have extracted and plan to use in several classifiers:

   - Word positivity/negativity - We have extracted this feature in two ways. The first involves counting words in 5-star reviews and words in 1-star reviews. We also count words in all reviews and remove the most frequent words from the 5-star and 1-star counts (to account for frequently-used words regardless of rating). Then the words are assigned weights based on their relative frequencies in each of the review types – positive for 5-star reviews and negative for 1-star reviews. These are then used to give a total "weight" to a review based on the words used in it. The other method of extraction is through SentiWordNet, a lexical database in which each word has an associated positivity, negativity, and neutrality score.
   - Punctuation - We have extracted the amount of certain types of punctuation used in a review. The idea is that more polarized reviews use more exclamation points and question marks. The lack of punctuation can also indicate a frenetic reviewer

giving a poor review. We also count quotation marks, as this can indicate sarcasm, and ultimately, a poor review.

- Capital letters - We have extracted the amount of capitalization used within a word. As with punctuation, people may tend to use all caps to express a strong emotion, indicating a polarized review (either 1 star or 5 stars). We only count capital letters within a single word, as counting all capital letters would introduce noise (proper nouns, capitalization rules, etc.).
- Consecutive repeated letters - As with capital letters and punctuation, it is common to repeat a letter consecutively many times to emphasize a word (e.g., *yummmmm* or *baaaaaad*). We again hope that this will be an indicator of polarized reviews.
- Time stamp - We have extracted several features from the time stamp of a review: whether the review was written in the morning (6am to noon), afternoon (noon to 6pm), or evening (6pm to midnight), the season in which it was written, whether or not the review was written on a weekend or national holiday (as opposed to a regular weekday), and the date itself. These features may reflect seasonal attitudes, time-of-day behaviors, and current trends (around a specific time period).
- Text length - As it implies, we also extracted the length of the text. Polarized reviews tend to take one of the extremes in terms of length.
- POS tags (specifically, adjective and adverb count) - We feel that polarized reviews may possibly use a greater number of adjectives than neutral reviews, so we have used a parser to tag sentences and extract the number of adjectives and adverbs used.

Changes

Current Goals

1. **Profile-Based Branch:**

    - We would like to train 4 classification models and compare and contrast the results obtained: Naive Bayes, Random Forest, Logistic Regression, and Support Vector Machines.
    - Using collaborative filtering, we would like to build a recommender system. It is item-based and user-based, and we intend to use matrix factorization.

2. **Text-Based Branch:**

    - We would like to train classifiers based on the features mentioned in this branch to get initial results.

- We would like to investigate which of the features are most useful and which are not as useful, as this can be a reflection of human time-sensitive behavior (time stamp features) or polarity expression (other text features).
- We would like to see if word positivity and negativity can be expanded to bigrams and trigrams, as well as parsing the sentence fully to instead look at whole noun phrases and verbal expressions.