Beer2Vec

Domain Background

The world of beer is expansive and the huge number of options can be intimidating for weekend Bud-drinkers and craft beer enthusiasts alike. Brews can range from a light lager to a creamy, nutty porter, and mouth-puckering sours challenge even the most seasoned of beer drinkers. For those seeking to find their next favorite drink or simply to expand their palettes some assistance navigating the universe of beers would be welcome.

Problem Statement

The challenge is to build a recommendation engine that can recommend a beer that matches the user’s tastes but may be new and undiscovered. The performance of such a model would be measured by using user reviews from a testing dataset. Each user has a set of beers which were rated positively and when given a subset of these beers as an input, the model output would be expected to include beers in the remaining set. This is a type of classification problem which takes input characteristics such as bitterness, sweetness, and alcohol content and assigns each beer to one of two labels, recommended or not recommended.

Datasets and Inputs

I plan to scrape www.beeradvocate.com to gather reviews from the 200 most prolific reviewers to represent both various flavor profiles and tastes. The features of interest will be 1) a unique user identifier e.g. userId 2) unique beer identifier e.g. name and brewery 3) the numeric score between 0-5 given each beer by the user. An example of the data gathered from a single user’s activity is shown below.

```
{ userId: ’morebeer’,
  reviews: {
        ’Tasty Caramel | Best Brewery’ : 4.7,
        ’Ok Beer | Bob’s Beers’: 3.3,
        ’Salty Stout | Mountain Hut’: 4.1
  }
}
```

I am defining a ’prolific reviewer’ as a user who has submitted 1000 - 10,000 distinct reviews. By scraping the history of 200 of these users roughly 1,000,000 data points will be collected.

Solution and Project Design

The desired solution to this problem is for the algorithm to return for a particular user a list of 10 beers which were not included in the user’s history of preferences but which has been classified as similar enough to the input to be recommended. From the beeradvocate.com dataset, users who have reviewed less than 1,000 beers will be used as the test dataset.

One way to build a recommendation system is content-based filtering, which uses the metadata around items to calculate similarity and thus make recommendations about new items[1]. Collaborative filtering on the other hand is based on a pool of past user actions, such as a previous reviewer’s history of preferences [1]. The downsides of collaborative filtering is that it requires user history, and may not perform well for users with uncommon preferences.

On the other hand, collaborative filtering can be used for datasets with little metadata and no domain knowledge of the items being recommended. A whiskey recommender was built using similar principles [2] using data scraped from reddit and applying the word2vec algorithm.

Word2Vec is an algorithm which was intially developed to create word embeddings to represent words from a large text input. The embedding captures semantics of words or ideas based on their proximity to other words or items in the input. After generating a vector space of beers based on reviews from beeradvocate.com, the space can then be used to find beers which are similar to a user’s known preference based on hidden parameters that the algorithm calculates.

Each reviewer’s history of reviews will be curated to select positive reviews (4+). These histories will be used as the input to an algorithm called Word2Vec which creates a vector space based on inferred features of each beer. To use the model generated by word2vec, user vectors will be generated based on an example user’s list of highly rated beers. The 10 beers with the highest cosine similarity to the user’s vector will be recommended.

Evaluation Metrics

The final model will be evaluated by curating positive (¿= 4.0) reviews from each user in the test dataset. The set should also be filtered to make sure each user has at least 100 positive beer reviews in their history.

```
{userId: ’testSubject1’,
 positive_reviews: {
        ’Dark | AB’: 5.0,
        ’Light | Coastal Creamery’: 4.5,
        ’Sour | Nice Place’: 4.9
        ...

 }}
```

Given a test user above, 10 beers from the user’s preferred list will be withheld to use as the testing dataset. The remaining beers will be input into the Word2Vec based algorithm and the success of the model will be evaluated based on how accurately its recommendations match the test dataset, that is what percentage of the 10 recommended beers are contained in the training set.

Benchmark Model

A similar model has been built and is available at https://www.recommend.beer. The method used by this recommendation system is based on creating a collaborative filtering matrix. The evaluation metric above can be used to compare the results of this model to the model proposed here by using the same testing datasets as inputs to the website recommender and evaluating the accuracy of its prediction.

1 Citations

1. Gong, S. (2011). A Personalized Recommendation Algorithm on Integration of Item Semantic Similarity and Item Rating Similarity. Journal of Computers, 6(5). doi:10.4304/jcp.6.5.1047-1054

2. Krzus, M. (n.d.). Retrieved December 04, 2017, from http://wrec.herokuapp.com/methodology

3. Z. (n.d.). Zackthoutt/wine-deep-learning. Retrieved December 04, 2017, from https://github.com/zackthoutt/wine-deep-learning

4. Itoku, E. (n.d.). What is your new favorite beer? Retrieved December 04, 2017, from http://www.recommend.beer/