Recipe Odorant Overlaps and Chef Classification Project Proposal

By Ali Sharman

Proposal Motivation and Summary

I have a strong interest in cooking and flavor combinations and I am studying information analysis and retrieval. The motivation for this project, therefore, is an attempt to merge these two interests by applying data science to recipe and flavor data. Furthermore, Food Network has a large amount of relatively easy to scrape data and flavor net is an interesting (and free) database of smells. Thus, the ability to play with some interesting datasets that will allow me to explore such concepts as similarity metrics, clustering algorithms, and network analysis is also a motivating factor. I also think that other people get excited about food and will, I hope, find the results and exploration of this data interesting.

To summarize the exploration, I hope to uncover the relationship between the odorants directly associated to ingredients through language (rather than chemically).[1] The main dimensions of the recipe-odorant relation that I am focusing on are odorant and odorant bigram, trigram popularity; odorant overlap within recipes; and chef classification by odorants compared to by categories. A more detailed explanation of the data and proposed analyses follows.

Datasets

Food Network

The Food Network is a popular television network that hosts cooking related shows. The associated Foodnetwork.com is a nicely organized website with underlying data that is easily scrapable. The website is organized by recipe and I would extract the following information for each recipe:

- Recipe title
- Chef name
- Ingredients

[1] There have been recent publications related to food pairing based on the actual flavor compounds found in different foods (See https://www.nature.com/articles/srep00196), but none, to my knowledge, based only on compounds in odors that are named after foods. The method based on actual flavor compounds is useful but requires the use of proprietary data. Beyond the issue of data, I am interested in keeping the language association filter in order to highlight shared properties between the most dominate/conceptually obvious flavors of foods.

- Categories
- Average review
- Number of reviews

Flavornet

Flavornet is a free database of human detectable odors based on Gas Chromatography Olfactometry (GCO). The database lists odors and their associated odorants. There are currently 738 odorants listed.

For example:

Odor	Odorant
banana	isobutyl acetate; isoamyl acetate; hexenyl acetate pentyl butanoate

Manipulating/joining the datasets

First, I will scrape the Food Network database and build a recipe json file to load into python. This file will contain the categories that I listed above.

Then, I will make a delimited text file of the odor, odorant data that I will take from the flavornet website and I will load that into python. I will also transform this into a dictionary with odor as the key.

Using spark, I will map the recipes to the odorants based on what odors are directly present as ingredients in the recipe. To do this, I will be matching the odor word to any matching word in the recipe ingredient list so that (applesauce, for instance, includes apple as an ingredient). Note: I will be tinkering with cleaning up this matching process so that I avoid mismatches such as pineapple.

I will take this matched dataset and register it as a sparksql table. I will exclude recipes that have no matching odors. The table will list all odorant-recipe matches and have the following fields:

ID, odorant, recipe name, category, chef, average rating

Computational tasks

Question 0: (warm up) Which odorants are associated with the highest reviews, on average?

I will use sparksql to obtain the average ranking of odorants based on the reviews that they appear in. I will then list the top 10 and bottom 10 of these odorants and the corresponding odors that they appear in. I will do the same with odorant bigrams and odorant trigrams based on counts in recipes.

Question 1: What is the degree of overlap in odorants within Food Network recipes?

I will use sparksql to count the instances of each odorant by recipe. I will then use spark to map recipes to a metric of overlap which I will calculate using a vector similarity measure (likely cosine) which I will apply to odorants listed within recipes and applying tf-idf weights obtained from the corpus of recipe odorants and finally, I will use spark to reduce the data to yield a recipe, odorant-overlap-measure output.

Question 2: Does a high degree of odorant overlap yield better or worse reviews than a low degree?

I will take the recipe, odorant-overlap-measure output and make into a sparksql table. Then, using sparksql, I will merge this table with the recipe table and to obtain the average review associated with the recipe. Then, using spark, I will create four histograms of odorant-overlap-measures with counts of all reviews, and then, of average review <3 stars, 3-4 stars, and 5 stars, respectively.

Question 3: Do some chefs prefer certain odorants? Do categories correspond to certain odorants?

I will return to the original odorant table. Here, I will use spark (map, reduce) to get a count of the number of times each chef uses each odorant. After eliminating any odorants that are very common to all chefs (the “stop words” of odorants) and normalizing odorant counts by the number of total odorants used in recipes of each chef, I will perform a cluster analysis to see if chefs can be grouped by odorant preference.

I will perform the same exercise but replace odorants with categories.

I will then compare the results of these analyses and see if the groups of chefs correspond.

Visualizations

1. A circular visualization of odorant data based on ratios in recipes to obtain a bird’s-eye-view of the scale of the data and the relationship between odorants within the data
2. Histograms as described under question2 to see recipe ranking by degree of similarity
3. Chef networks to compare with each other
    i. ties based on odorants, directional weighted by counts, color by classified group
    ii. ties based on categories, directional weighted by counts, color by classified group