

John Zhang, jwz2kn@virginia.edu
Tyler Ealy, tre7ca@virginia.edu
CS 4710 Artificial Intelligence
04/26/2016

Homework 4 Writeup

Note for running the program properly: There are 3 paths that you must change. You must give the program your path for the ingredients.txt, training_data.json, and whatever testing .json you choose to use.

We used the Naïve Bayes algorithm to determine the most likely cuisine of a recipe given the ingredients of that recipe. We followed the algorithm as it was described in class, computing:

$$V_{nb} = \underset{v_j \in V}{\operatorname{argmax}} \widehat{Pr}(v_j) * \prod_{i=1}^n \Pr(a_i | v_j)$$

In that equation, a_i is the ingredient given the recipe, and v_j is the cuisine type. The result V_{nb} is the cuisine with the highest probability of being the correct label for a given recipe.

We made a couple of changes to our basic implementation of the algorithm in order to optimize it. In one section there was a triple “for” loop where the program searched through every single ingredient inside a loop that went through the entire list of ingredients. This was a huge amount of searching that could be reduced by slightly increasing the space complexity. We optimized this search by creating a dictionary where the keys were the 20 cuisines, and values were lists containing all recipes of each cuisine. We had the program search through this list when looking for ingredients inside the first two “for” loops so that it was not necessary to search through every single recipe every time. We also added a simple bit of rote learning to our algorithm. If a testing recipe is the exact same as a recipe we used in training, it will always return the correct cuisine type.

This combination of rote learning and Naïve Bayes classification gave us an algorithm that spent around 0.6 seconds and 14 MB of memory to learn for the given data set. These specs were done using a 2015 Macbook with an Intel Core i7 2.2 GHz with the latest version of OSX, running no other programs other than Google Chrome and PyCharm. These are printed within the program.