ISE 533: Integrative Analytics, Spring 2020

Meal Planning Draft

Group 1:

Xinjie Sun

Xinrui Yan

Yiwen Cao

Zelong Chen

Abstract

The project is opted to design a user-friendly meal plan recommendation system that aims to help two individuals plan out their dinner for five days and allocate cooking time based on their combined schedule. To achieve the objective, we first generate user profiles containing the users' health condition, eating preference, schedule, and spending habits from user inputs. Through collaborative filtering, our recommendation system is able to use the user profiles to predict the users' rating on recipes they have not rated based on past user's rating history. Using these prediction ratings and the user profiles, our optimization model will look to maximize the sum of both user ratings while ensuring the users' nutritional requirements are fulfilled, the budget is contained, the food allergies are avoided, the cooking time is within the available time for the user, and the cooking time of each one in the household is similar. The system will then output a meal plan with different recipes which users are most likely to enjoy, along with information on whose is cooking each day, cook time, and ingredients needed. We are able to generate different meal plans which sufficiently satisfy the users' preference. Because our personalized meal plan could suit more individuals compared to the traditional recipe book and the self-generating process could be more timesaving for people with a busy lifestyle, our application could be a better alternative to the traditional meal planning.

Goal and Scope of Project

This project focuses on the design of a meal plan recommendation system dedicated to the New Millennium. The generated meal plan from the recommendation system is opted to suit the lifestyle and the preference of the users. In order to develop the recommendation system and the optimization model, the group lies the following general assumptions.

Assumptions

- The meal plan is designed to accommodate a household of two with one dinner meal a day, and five days a week.
- In order to adapt the personal spending habits and schedule, the team decides to let the user set the budget of ingredients and cooking time.
- To generate nutritional requirements, we first assume each dinner meal will contain 40% of the daily nutritional intake. Then the program uses information such as the users height, weight and age to calculate the daily nutritional requirements that were laid out by the FDA guidelines

- To calibrate the meal plan regarding personal health conditions, the group designs the program to take the users' allergies into consideration.
- To accommodate the schedule in the household, the group assumes both persons take about the same time to cook every week.
- To best simulate the reality, the group decides to eliminate spices from the recipe to avoid additional cost. And the team distinguishes the minor and major ingredients based on the measuring unit.
- The team provides three purchasing options to the user to accommodate different purchasing habits.

Overview of Models/Algorithms

The meal plan recommendation system is mainly composed of two parts, the Recommender System and the Optimization Model. The logics the group applies to the model are as follow:

Recommender System (Collaborative Filtering)

- Using the user ratings that were scraped from the online source, we can generate a characteristic matrix containing each user and their ratings on recipes that they have tried before. We can expect the characteristic matrix to be very sparse as users will only be able to try and rate a small number of these recipes from the large collection recipes on the website. Thus, we decided to merge every 50 users into one to reduce the amount of sparsity within the characteristic matrix.
- To get an initial idea of the users' taste preference, the system will output ten random recipes for each user to rate. In addition, food allergies are also taken into consideration where recipes containing these ingredients will be rated with a score of zero. The system will then concatenate the current users' rating with the characteristic matrix generated from the previous step. Then using collaborative filtering with methods such as Nuclear Norm Minimization (NNM) or Soft Impute (Singular Value Decomposition) from the Python Fancyimpute library, the characteristic matrix can be completed. Thus, generating the predicted ratings of all recipes for our current users.
- Fancyimpute
  - Source: https://github.com/iskandr/fancyimpute
  - Nuclear Norm Minimization (NNM)
    - Matrix Completion using convex optimization to find low-rank solution

        - Simple implementation of ***Exact Matrix Completion via Convex Optimization*** by Emmanuel Candes and Benjamin Recht using *cvxpy*.
    - Soft Impute (Singular Value Decomposition)
        - Instead of solving the nuclear norm objective directly, instead induce sparsity using singular value thresholding
        - Matrix completion by iterative soft thresholding of SVD decompositions. Inspired by the softImpute package for R, which is based on *Spectral Regularization Algorithms for Learning Large Incomplete Matrices* by Mazumder et. al.

Optimization Model

- The Optimization problem is opted to get a best satisfied meal plan for a household of two. The objective function is designed to get the meal plan with best simulated ratings while satisfying all the assumptions the group lies. In addition, unequal cooking time will lead to penalty in the objection function so that the group keeps the cooking time of partners as close as possible. The nutrition requirements are calculated based on the body information given by the user, and the user inputted information is used to fill in the parameters.
- Mathematical Formulation

Indices:

$$Person: i \in \{1,2\}$$
$$Day: t \in \{1,2,3,4,5\}$$
$$Recipes: k \in \{1,2,\ldots,K\}$$

Parameters:

$$Available\ Time: T_{it}$$
$$Budget: B$$
$$Recipe\ price: \pi_k$$
$$Nutrition\ Vector: A_k = (cl_k, s_k, f_k, p_k, cb_k, fi_k)^T$$
$$Minimum\ Requirements: R_l = (cl_k, s_k, f_k, p_k, cb_k, fi_k)^T$$
$$Maximum\ Requirements: R_u = (cl_k, s_k, f_k,\ p_k, cb_k,\ fi_k)^T$$
$$Cooking\ Time: \tau_k$$
$$Ratings\ for\ each\ recipe: r_{ik}$$
$$Penalty\ for\ inequity: \alpha$$

Variables:

$$x_{itk} \in \{0,1\}\ indicate\ whether\ person\ i\ cook\ recipe\ k\ on\ day\ t$$
$$w: inequity\ cooking\ time$$

Optimization Problem:

$$\text{Max} \sum_{j=1}^{K} \sum_{t=1}^{5} \sum_{i=1}^{2} (r_{1k} + r_{2k}) x_{itk} - \alpha w$$
$$\sum_{j=1}^{K} \sum_{i=1}^{2} x_{itk} = 1. \forall t \in \{1,\ldots,5\}$$

$$\sum_{t=1}^{5} \sum_{i=1}^{2} x_{itk} \leq 1. \forall k \in \{1, \dots, K\}$$

$$\sum_{j=1}^{K} \sum_{t=1}^{5} \sum_{i=1}^{2} A_k x_{itk} \geq R_l$$

$$\sum_{j=1}^{K} \sum_{t=1}^{5} \sum_{i=1}^{2} A_k x_{itk} \leq R_u$$

$$\sum_{k=1}^{K} \sum_{t=1}^{5} (x_{1tk} - x_{2tk})\tau_k \leq w$$

$$\sum_{k=1}^{K} \sum_{t=1}^{5} (x_{1tk} - x_{2tk})\tau_k \geq -w$$

$$\sum_{j=1}^{K} \sum_{t=1}^{5} \sum_{i=1}^{2} \pi_k x_{itk} \leq B$$

$$\sum_{k=1}^{K} x_{itk}\tau_k \leq T_{it}, \forall i \in \{1, 2\}, t \in \{1, \dots, 5\}$$

$$x_{itk} \in \{0, 1\}, i \in \{1, 2\}, t \in \{1, \dots, 5\}, k \in \{1, \dots, K\}$$

Data Sources and Data Processing

For Data Collection, we used a web scraping library called BeautifulSoup to collect various data from several websites such as:

- Yummly.com
  - Food recipes contain nutritional information, user ratings, ingredients, and cook time.
- Grocery Websites (Amazon, Walmart, Wholefoods)
  - Cost of Ingredients
- Fda.gov
  - Daily Nutritional Requirements

In the process of data cleaning, we obtained all available recipes from Yummly.com and observed the dataset. Since the dataset is large enough to eliminate unnecessary recipes, the group dropped all recipes with NAs and recipes with less than four reviews with 265 recipes left. Then, we obtained information regarding nutrition information, ingredients, cooking time. To better prepare for the recommendation system, we created the rating matrix containing all existing ratings for recipes, and for the ease of computation, we group 50 users' rating into 1 user's to avoid great sparse space.

In the process of price scraping, we use the ingredients list obtained from the recipe as search keys to create search urls. We created the algorithm that replaces each ingredient into the search url to obtain prices of each ingredient. And we saved prices from different sites into different dataset to prepare for the optimization. In the Optimization output, the different options are given.

In regard to the Nutritional requirements, we obtained the nutritional factors' bound from the FDA website, and implemented the bound calculation into our code, which intakes the user's body information, calculates the ratio based on height and weight, and uses the ratio to time the standard bound to get the advised nutritional intake.

Discussion of Results

Current result

The system we computed is able to give a meal plan with five recipes with simulated highest ratings considering the users' height, weight, eating habits, and spending habits. The output meal plan contains the ingredients needed, cooking time for each recipe, the person who is in charge of cooking on each day and prices from three sources which the user could choose by their preference (Walmart, Wholefoods, and Amazon). In addition, the output meal plan takes the height, weight, and allergies for each user, therefore, the daily nutritional requirements are met, and no allergies food would appear in this output to ensure the user experience. Considering the lifestyle of roommates, the output simulates recipes that allow users spending similar time on cooking each week since we believe equal responsibility in cooking meals could improve the user experience. We achieved this by adding penalty cost in our Optimization problem if unequal cooking time existed.

Discussion on the Alternative Parameters and Alternative Scenarios

Our system could take different users' information and generate personalized meal plans based on the input. Since we have 265 recipes, we could fit in different budgets and eating allergies, however, our system is less sufficient in adapting to the varying time that users are willing to spend on cooking. All of our recipes require longer preparation time of 30 minutes or longer, thus, the meal plan might be less realistic for busy lifestyles. In addition, our system could implement more personalized choices, such as more between meal snacks and fruit options to better suffice the nutritional requirements, diet restrictions, or pick eaters.

The two different methods of implementation in our recommendation system varied in performance. With different trials, it seems that Nuclear Norm Minimization (NNM) produced a better result in predicted rating accuracy. We believed the difference in result from Soft Impute (Singular Value Decomposition) arised from sparsity within the characteristics matrix. Without more user rating data, the NNM method is prefered over the Soft Impute (SVD) even though the computation time is much longer.

Future Work

Recommender System

- Improve on our current recommender system by implementing a hybrid approach consisting of both User-based Collaborative Filtering and Content-based recommender
    - User-based Collaborative Filtering uses user's past behavior and similar decisions made by other users to predict recipes that the user may like
    - Content-based recommender uses the characteristics of an recipe to recommend similar content to the recipes the user liked in the past

Optimization Model

- Adding more flexibility to the model by introducing penalty rather than directly using hard constraints.
- Giving users more personalized lifestyle choices rather than simply calculate the nutritional requirements based on users' height and weight. For instance, one of the users plans to lose weight, then he may need less calories.

User Interface

- Creating a user-friendly interface to enter their height, weight, budget, allergies and their rating for 10 random recipes.
- Collecting user's feedback and improve the accuracy of the recommender system according to users' ratings on the output recipes.