

# A Network Tour of Kitchen Science: Graph-Based Ingredient Suggestion and Replacement

Erick Maraz, Lara Orlandic, Niccolò Stefanini, Valentin Viennot

**Abstract**—When cooking, people often find it necessary to replace an ingredient due to intolerance or personal taste. Or sometimes, one may want to add a new taste to a known recipe. In order to help amateur chefs accomplish these tasks, we use data science and graph manipulation to develop an ingredient suggestion and replacement tool. We first develop several ingredient networks using the Recipe1M+ and USDA databases. The properties of each network are analyzed and the graphs are combined to take into account both taste coherence and nutritional information when developing new recipes. Subsequently, community detection and ingredient hub detection are performed, and three different graph filtering methods are implemented on ingredient signals and their output recommendations are analyzed to qualitatively assess the success of each ingredient suggestion algorithm. Finally, we use a combination of nearest neighbors analysis and graph filtering to determine the optimal replacements for ingredients, successfully developing palatable choices both for meat-loving and vegan users.

## I. INTRODUCTION

Adopting a vegetable-based diet can greatly reduce an individual’s carbon footprint [1]. However, replacing meat products with plant-based ones can be challenging. Similarly, finding the adequate replacements for ingredients or additions to recipes is difficult for people who are not expert chefs.

We propose a tool to remove certain ingredients from a recipe and output the best ingredients to replace them, considering both nutritional and recipe context information, and to suggest ingredient additions.

First, we explain how we curate the data and build different ingredient graphs. We compare our graphs’ properties and gain some insights on their structures (i.e. hubs, communities). Then, we explore graph filters and other network manipulation tools to perform label propagation, which produces ingredient suggestions, as well as ingredient similarity comparison for ingredient replacement. Finally, we propose a method producing coherent ingredient replacement results in terms of both nutritional

properties and consistency with the taste of the recipe.

## II. DATA ACQUISITION AND EXPLORATION

Intuitively, there are two components to successful ingredient suggestion and replacement: an ingredient must contain similar nutritional facts as the one it is replacing, and it must have a complimentary taste to the other ingredients in the recipe. Therefore, we utilize data indicating which ingredients are commonly used together in recipes, as well as the nutrition facts of those ingredients.

### A. Recipe1M+ Database

We use a subset of the Recipe1M+ database containing ingredient nutrition information [2]. This dataset consists of over 50,000 recipes, along with their ingredients, nutrition facts per ingredient, preparation instructions, and health scores.

We originally planned to use both the ingredient lists and the nutrition facts per ingredient, to develop adjacency matrices in which nodes represent ingredients. However, we realized that the nutrition facts per ingredient were scaled to the ingredient quantity, and that this data was not consistent across different measurement units. For example, 10 oz of broccoli are less than 2 cups, but 10 cups are reported to contain 34% more fat than 2 cups. Therefore, we needed to merge this data with another database to obtain accurate nutritional information for each ingredient.

### B. USDA Nutrition Information Database

We used the USDA Nutrition Information Database to determine the nutrition facts of each ingredient found in the Recipe1M+ database [3]. When the full ingredient could not be matched to one in the USDA database, the first word of the ingredient was matched to the first word of

an ingredient in the USDA database. Overall, 313 of the 357 unique ingredients in the Recipe1M+ database could be located in the USDA database.

### C. Adjacency Matrix Creation

Using the data from the aforementioned databases, we developed 12 adjacency matrices to experiment with different graph creation techniques. In each adjacency matrix the nodes represent ingredients, but the edge weight calculations differ. The properties of each graph were then analyzed, and the different graphs were tested on our ingredient suggestion and replacement algorithms to see which matrices produce valid results. The graphs used in the final solution are described as follows.

*1) Recipe Co-occurrence Adjacency Matrix:* In the first adjacency matrix we created, the weights between ingredient nodes are defined by the number of recipes in which the two ingredients appear together in the Recipe1M+ dataset. These weights range from 0 to 8385, and the two ingredients most frequently used together were salt and white flour. This graph is fully-connected with a diameter of 3 and a clustering coefficient of 0.75. A histogram of its degree distribution is shown in Fig. 1. We can see that the graph resembles that of a random network, as the degree counts of its nodes approximately follow a Poisson distribution.

*2) Nutrition Adjacency Matrix:* Next, we create a feature matrix for each ingredient found both in the Recipe1M+ and USDA databases. The features are the values of 10 common nutrients, including protein, total fat, and carbohydrates. Each nutrient column is then divided by its mean value to ensure that nutrients with naturally higher quantities are not weighted more heavily than others. Then, a Radial Basis Function (RBF) kernel, described in Eq. 1, is applied to compute the weights between ingredient feature vectors  $x$  and  $x'$ .

$$W_{x,x'} = e^{-\frac{\|x-x'\|^2}{2\sigma^2}} \quad (1)$$

$\|x - x'\|^2$  represents the Euclidean distance between feature vectors, and we set  $\sigma = 0.8 * \text{mean}_{\text{distance}}$ , meaning 80% of the mean Euclidean distance. Finally, the matrix was sparsified by setting all weights lower than  $\epsilon = 0.2$  to 0. These parameters were selected to have a fairly constant weight distribution between 0 and 1. This network

is disconnected with 6 components and a clustering coefficient of 0.94, meaning that the nodes that are connected are highly clustered together. Its degree distribution is plotted in Fig. 1, and we can see that it follows a sort of reversed power law distribution, where few nodes have low degrees and many nodes have high degrees.

*3) Combined Recipe and Nutrition Adjacency Matrix:* In an attempt to take both nutrient and recipe co-occurrence information into account in our network, we experimented with different adjacency matrix combination techniques. For example, we took the aforementioned co-occurrence adjacency matrix and divided it by its maximum value so that all of the entries would be from 0 to 1. Then, we added it to the nutrition adjacency matrix so that the weights would represent both co-occurrence of ingredients and nutritional similarity. This graph is fully-connected with a diameter of 2 and clustering coefficient of 0.94. Its degree distribution histogram is shown in Fig. 1 and also follows a reversed power law distribution. Other adjacency matrix combination techniques we tested were multiplying the two matrices and concatenating nutrient features with recipe "features" (booleans representing whether or not an ingredient is present in a recipe) and applying an RBF kernel as in Eq. 1.

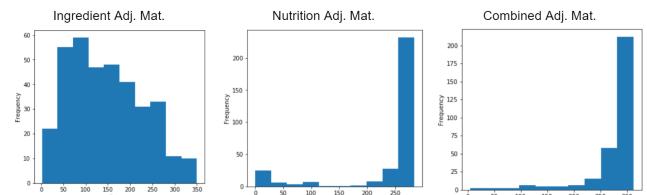


Fig. 1: The degree distribution histograms of the recipe co-occurrence matrix, the nutrient adjacency matrix, and the sum of the two.

## III. INGREDIENT NETWORK EXPLORATION

### A. Ingredient Hub Detection

First, we use the notion of degree centrality – the fraction of all nodes to which a given node is connected – to find ingredient "hubs," which are ingredients that are very commonly found in recipes. We define these hubs as nodes with a centrality higher than 2 standard deviations above the average centrality on the recipe co-occurrence adjacency matrix described in Section II-C1.

We found the following hubs: butter, honey, raw lemon juice, 1% fat milk, olive oil, salt garlic powder, pepper, sugar, water, and wheat flour. These results make sense, as these ingredients are kitchen staples.

### B. Community Detection

An intuitive outcome for the ingredient graph would be to have distinct clusters of similar ingredients, such as salty and sweet ingredients. Community detection performs such clustering based on high average degree subgraphs. We use community detection on the three aforementioned ingredient networks.

To observe if our previously built graphs show some ingredient communities, we utilize the Louvain algorithm [4]. This algorithm determines the graph partition that produces the highest graph modularity. The algorithm proposes an heuristic method that can be applied to big graphs.

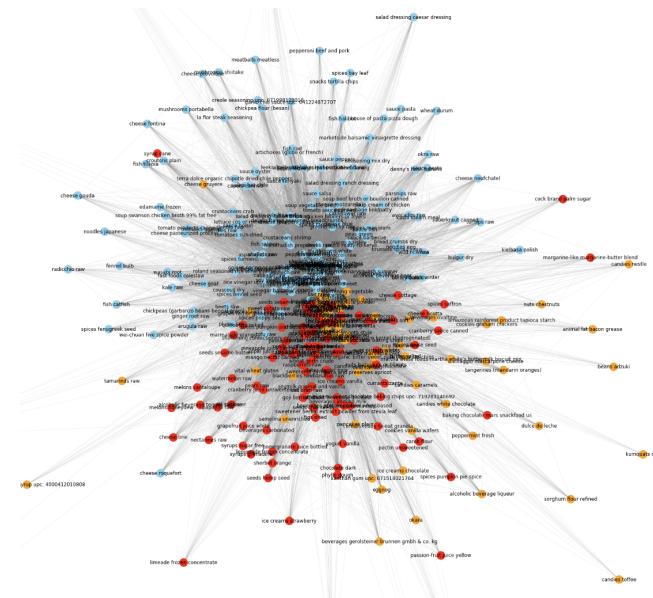


Fig. 2: Community detection on the recipe co-occurrence network.

On the graph built from recipe co-occurrence adjacency matrix, displayed in Fig. 2, two large communities tend to appear: salt (blue) vs. sweet (red) and fat/sweet (orange).

On the graph built from nutrition adjacency matrix, depicted in Fig. 3, communities are much more clearly defined: sweets (red), fats and proteins (orange), meats and cheeses (turquoise), and multiple spice clusters.

Finally, meaningful clusters could not be identified from the combined adjacency matrix. From these graph visualizations, we hypothesize that the nutritional graph will yield good results for replacing ingredients with a similar role in the recipe, since such ingredients are clustered together, whereas the ingredient co-occurrence graph will give better suggestions for ingredients to use in common.

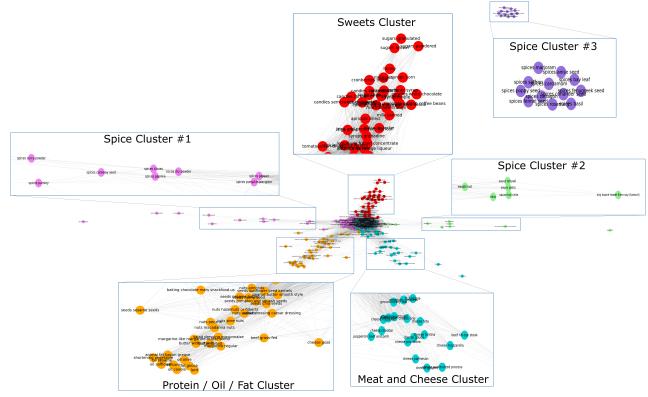


Fig. 3: Community detection on the nutrition network. Distinct communities are visible.

## IV. GRAPH FILTERING FOR INGREDIENT SUGGESTION

We propose a method of ingredient suggestion in which a recipe signal is transformed using the ingredient graph to magnify other ingredients that would fit well in the recipe. The recipe signal is a boolean signal with values of 1 at the indices of ingredients that the recipe contains. The suggested ingredient is defined as the ingredient that was not in the original recipe but whose index has the highest value in the transformed signal. We try three different transformation algorithms, all of which utilize knowledge of the ingredient network, and test them using the ingredient co-occurrence adjacency matrix.

First, the graph shift algorithm described in [5] was implemented using the recipe feature adjacency matrix. In this algorithm, the signal,  $s$ , is simply multiplied by the adjacency matrix,  $A$ , as shown in Eq. 2.

$$\hat{s} = As \quad (2)$$

Theoretically, this algorithm expresses each signal value as a linear combination of its neighbors.

In practice, however, this method tends to repeatedly suggest the same ingredients that have large weights, such as salt and pepper.

Next, we try the Graph Total Variance (GTV) minimization method described in [5]. In this algorithm, all possible versions of the original recipe signal, denoted  $\tilde{s}$ , are generated by adding a 1 at indices where the ingredient was not previously present. Then, the optimal value is defined as the signal that minimizes the GTV, as shown in Eq. 3

$$\hat{s} = \underset{\tilde{s}}{\operatorname{argmin}} \tilde{s}^T L \tilde{s} \quad (3)$$

where  $L$  is the Laplacian of the adjacency matrix. Since this method aims to minimize the variation on the graph, it also outputs the same common ingredients (e.g. butter, salt) repeatedly.

Finally, we implement a graph bandpass filter on the signal to perform ingredient suggestion. A bandpass filter was selected because a lowpass filter would minimize variation on the network and therefore output only very common ingredients (e.g. salt, pepper), whereas a highpass filter might output very exotic ingredients (e.g. kumquats) that do not work well with the recipe. To determine the passband of the filter, we plotted several ingredient signals, and found out that the greatest signal activity occurs between  $0.8 \leq \lambda \leq 1.1$ , where  $\lambda$  represents an eigenvalue of the adjacency matrix. This can be shown in Fig. 4, which shows the Fourier Transform of an ingredient signal. Therefore, we selected [0.8, 1.1] as the passband of our filter.

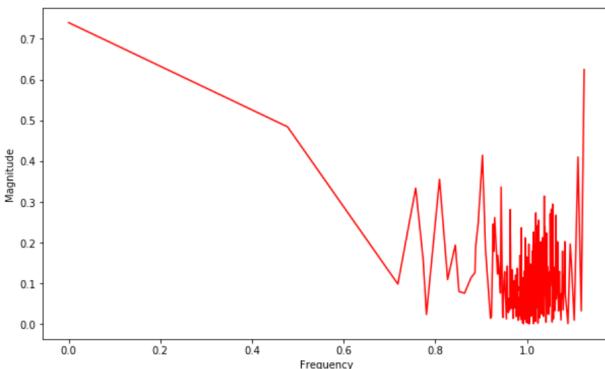


Fig. 4: A graph Fourier Transform of the recipe signal representing sugar cookies.

The graph filter is applied by performing a graph Fourier Transform on the signal to convert it into the frequency domain, and then multiplying it by the ideal bandpass filter. This signal is then transformed

back into the time domain, and the new ingredient is detected. This approach is shown to work relatively well; it suggests non-obvious ingredients that respect sweet/salty food conventions. For example, it suggests adding hash browns to chicken casserole and vanilla extract to chocolate-orange angel food cake.

## V. kNN INGREDIENT REPLACEMENT

Suggesting new ingredients based on which ingredients are commonly used together is useful, but sometimes one must remove an ingredient. In this case, it is necessary to incorporate nutritional information, since the new ingredient should provide the same nutritional value as the one it is replacing. To perform this task, we replace each ingredient with its  $k$  Nearest Neighbors, which are the nodes to which it is connected that have the highest edge weights, on the nutrient adjacency matrix. This solution produces promising results: it suggests replacing beef with catfish and almond milk with ovaltine.

### *Vegan Ingredient Replacement*

This kNN ingredient replacement can be used to help vegans remove animal products from recipes by finding products that offer similar nutritional benefits. We implemented this "veganization" of ingredients by first determining which ingredients contain Vitamin B-12, which is only found in animal products. These ingredients' adjacency weights are then set to zero, effectively removing them as neighbors, and then the algorithm is performed to find the best replacements. This approach produces promising results for meat products (ex. replacing beef with edamame or tofu). However, it does not work well for milk products (ex. replacing brie cheese with peanut butter) due to the fact that there are only 226 vegan ingredients in the database, many of which are vegetables or spices and not milk or cheese replacements.

## VI. FINAL RESULTS

Graph filtering label propagation on the ingredient co-occurrence matrix produced ingredient suggestions that were consistent with the recipes, but could not be used for ingredient replacement as the nutritional information of the removed ingredient was missing.

On the other hand, kNN gave consistent ingredients suggestions in terms of nutritional properties and could be used to find alternatives. However, it was missing contextual information about the input recipe and therefore the expected taste output.

We propose a combination of both of these methods for an efficient and innovative “ingredient replacer” using both removed ingredient properties and recipe context:

- We project the recipe signal over the recipe co-occurrence network.
- Then, we apply the ideal filter found in Section IV and save the obtained signal for later use.
- We apply the kNN ingredient replacement on the removed ingredient to find the k closest ingredients on the nutrition graph (which can be “veganized” depending on user preferences).
- Finally, we rank these k ingredients by their previously obtained signal from highest to lowest. This is the preferred suggestion order for the given recipe.

Table I shows the results obtained by applying this algorithm on a broccoli chicken casserole. The ingredients are replaced one-by-one and displayed in bold.

TABLE I: Ingredient Replacement Results

Ingredients: broccoli, cheddar cheese, chicken, lemon juice, white rice, mayonnaise salad dressing, cream of chicken soup			
<b>broccoli</b>	cauliflower	adzuki beans	winter squash
<b>cheddar</b>	roquefort	gouda	neufchatel
<b>chicken</b>	tofu	pork	eggs

We applied our algorithm to multiple recipes and observed similar replacement suggestions to those found on cooking blogs: honey for sugar, yogurt for sour cream, molasses for honey, and vinegar for lemon juice [6].

## VII. CONCLUSIONS

Using recipe co-occurrence and ingredient nutrition data, we studied the properties of different ingredient network graphs and performed community detection that successfully distinguished between different families of ingredients. Then, we studied the k Nearest Neighbors of the nutrition network in order to find the most nutritious replacements for ingredients, as well as their best vegan alternatives. This approach achieved moderate success given the limited number of vegan ingredients. Next, we

experimented with different ingredient and graph signal processing techniques and determined that a graph bandpass signal could be used to produce feasible ingredient suggestion for recipes. Finally, we combine the graph filtering and kNN approaches to produce a tool that learns from both the recipe co-occurrence and nutritional networks to suggest the best replacements for a given ingredient in a recipe, achieving promising results.

We were able to produce an ingredient replacement and recommendation system that outputs seemingly tasty recipes. However, our work is limited by the fact that there is no objective way to quantify “tastiness,” as people’s food combination preferences vary widely. Therefore, we were only able to provide qualitative assessments of which methods produced feasible recipes.

A further limitation is that the dataset we used only consisted of 357 unique ingredients, 226 of which were vegan, and most recipes were North American. Using a more diverse database with food across cultures may produce more interesting ingredient suggestion results.

Finally, our ingredient suggestion results are not perfect due to the fact that we did not take ingredient preparation or quantities into account. For example, our ingredient suggestion graph filter algorithm was inputted butter, parmesan cheese, sweet corn, and olive oil and outputted Alaskan king crab. This sounds like a tasty solution until you see that the original recipe was for “cheesy popcorn,” which may not go well with crab. Therefore, we should take a wider variety of features into account.

Some future pathways for this project would be to focus on a specific type of recipe (ex. cocktails, sauces, desserts) to refine the suggestion algorithms to a specific type of food. Furthermore, we could solve the issue of unobjective results by cross-referencing our generated recipes with larger databases of existing recipes to see whether our suggestions have been tried before. Next, we could consider the possibility of replacing one ingredient with a combination of others, as is common practice in cooking blogs [7]. Finally, we could train a GCNN to learn the structure of the ingredient graph and determine the optimal graph filter to use for our ingredient suggestion, where the data labels would be whether or not a produced recipe exists in a larger recipe database.

## REFERENCES

- [1] A. Rosi, P. Mena, N. Pellegrini, S. Turroni, E. Neviani, I. Ferrocino, R. Di Cagno, L. Ruini, R. Ciati, D. Angelino, J. Maddock, M. Gobbetti, F. Brighenti, D. Del Rio, and F. Scazzina, “Environmental impact of omnivorous, ovo-lacto-vegetarian, and vegan diet,” *Scientific Reports*, vol. 7, no. 1, 12 2017.
- [2] J. Marín, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, “1 Recipe1M + : A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images,” Tech. Rep. [Online]. Available: <https://www.internetworldstats.com/stats.htm>
- [3] “USDA Food Descriptions.” [Online]. Available: <https://www.ars.usda.gov/>
- [4] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, 10 2008.
- [5] A. Sandryhaila and J. M. Moura, “Classification via regularization on graphs,” in *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*, 2013, pp. 495–498.
- [6] “18 Most Common Ingredient Substitutions — Food Network Canada.” [Online]. Available: <https://www.foodnetwork.ca/kitchen-basics/photos/18-most-common-ingredient-substitutions/>!6c50e14466702b64250c278690049a3f
- [7] “Missing An Ingredient? This List Will Solve All Your Problems.” [Online]. Available: <https://www.onegoodthingbyjillee.com/common-ingredient-substitutions-list/>