Recipe Completion using Machine Learning Techniques

Marlies De Clercq MM.DECLERCQ@UGENT.BE
Michiel Stock MICHIEL.STOCK@UGENT.BE
Bernard De Baets BERNARD.DEBAETS@UGENT.BE
Willem Waegeman WILLEM.WAEGEMAN@UGENT.BE
KERMIT, Departement of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure links 653, 9000 Ghent, Belgium

Abstract

Completing a recipe is not trivial. The success of ingredient combinations depends on a lot of factors such as taste, smell, texture, etc. The aim of our work is to build a model that adds one or more ingredients to a given number of ingredients. The idea is based on leftover ingredients in a fridge. A person could list the available ingredients in his or her fridge and the model would suggest some ingredients to create a full recipe.

1. Introduction

There already exist several methods to complete a recipe. The first method is to look for existing recipes that contain one or several of the leftover ingredients. This is done using cook books or online search engines. Examples are supercook.com, myfridgefood.com and recipematcher.com. The second method to find some suitable ingredients to add to the remaining ingredients in the fridge, is to use computational models. One example is the online model of Foodpairing N.V., which gives for one ingredient those ingredients that make the best combination with the given ingredient. The model is based on the food pairing theory which states that two ingredients make a good combination when they have major flavour components in common. Unfortunately, neither of these two methods can tell which type of meat can best be combined with all remaining ingredients, or which herb makes the best combination with all ingredients present. Therefore, we have built two data-driven models that can solve such problems. These models give, for a given set of ingredients, those ingredients that can best be combined with all of the given ingredients. A first model applies non-negative matrix factorization and is restricted to using a database of existing recipes to gather information on ingredient combinations. A second model is based on the two-step recursive least squares method. This model bases its suggestions not only on existing recipes, but also on the flavour profiles of the ingredients. This makes it possible to find new ingredient combinations.

Short version of a submitted article, Copyright 2015 by the author(s).

2. Materials and Methods

The two data-driven models are built using data provided by Ahn et al. (2011). A first data file contains 56,498 recipes coming from eleven different cuisines. For each recipe the accompanying ingredients are enumerated. The total number of different ingredients is 381. This data file is transformed into a binary matrix as

$$Y_{ri} = \begin{cases} 1, & \text{if ingredient } i \text{ is present in recipe } r; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Only recipes with three or more ingredients are taken into account, reducing the number of recipes to 55,001. A second data file contains the names of 1,530 ingredients and their corresponding category (fruit, meat, herb, etc.). The third data file enumerates 1,107 flavour components that are found in foodstuffs. A fourth data file links ingredients with their flavour components. These three data files are combined into a second binary matrix as

$$X_{ic} = \begin{cases} 1, & \text{if flavor component } c \text{ is present in ingredient } i; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Only the 381 ingredients found in the recipes and their flavour components are taken into account, resulting in an $381 \times 1,021$ matrix X.

2.1. Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a decomposition technique that approximates a matrix by a product of two low-rank matrices. This results in the elimination of noise in the data. NMF assumes that the data is non-

negative and only allows additive combinations to represent the data. This leads to the unique part-based characteristic of the method: NMF represents the data by combining different parts of the learned data (Lee & Seung (1999)). All parts are used to represent at least one instance, but not all parts are used for each instance. This facilitates the interpretation of the representation of the data. It also means that matrix W and matrix H contain a lot of values that are zero, leaving out those parts that are not needed to represent a certain instance. As a result, the matrices are sparse. A given matrix is approximated as

$$Y \approx WH, \tag{3}$$

where Y is an $m \times n$ matrix, W is an $m \times k$ matrix and H is an $k \times n$ matrix. Parameter k is the rank of the factorization and determines the number of latent features. All entries of W, H and Y need to be non-negative.

2.2. Two-step recursive least squares

A difference between NMF and two-step RLS is that the latter not only uses information about ingredient combinations captured in the recipe matrix Y, but also adds information on flavour profiles of ingredients, gathered in matrix X. Two-step RLS differs from NMF, both methods represent data by multiplying matrices, however, the matrices of two-step RLS are not constructed by decomposing the original data. The equation of two-step RLS can be seen as

$$Y \approx K_u W K_v, \tag{4}$$

where K_u and K_v are two kernel matrices, which contain information that can help representing the data. W is a coefficient matrix that will be trained to minimize the error between Y and $K_u W K_v$. To prevent overfitting, the coefficient matrix is estimated as

$$W = (K_u + \lambda_u I)^{-1} Y (K_v + \lambda_v I)^{-1}, \tag{5}$$

using regularization. The optimal values of λ_u and λ_v, the hyperparameters, are determined during validation. More information on two-step RLS can be found in Pahikkala et al. (2014). In our model, matrix Y from two-step RLS is the binary recipe matrix (Y). K_u and K_v are linear kernels of respectively the recipe data (Y) and the flavour data (X). K_u represents the number of shared ingredients for each pair of recipes. K_v contains for each pair of ingredients the number of flavour components they have in common, allowing to add ingredients with shared flavor components (based on the food pairing theory) to complete the recipe. Two-step RLS can be used to complete new recipes, by adding ingredients of these new recipes to K_u; to add new ingredients to existing recipes, by adding flavour components of new ingredients to K_v; or to add new ingredients to new recipes. A graphical representation of these actions is given in Figure 1.

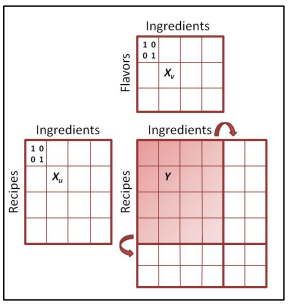

Figure 1. Graphical representation of two-step RLS applied to complete recipes.

3. Results

Canonical correlation analysis is applied on the recipe and flavour data. The linear distribution of these points confirms the relation between the flavour composition of ingredients and the way ingredients are combined in recipes (food pairing theory). Mussel, cognac, fig and star anise are examples of ingredients with a high correlation. Butter, onion, brown rice and egg on the other hand are ingredients with a low correlation between flavour components present and use in recipes.

To tune the models and test their performance of completing recipes, tune and test recipes are selected. For two-step RLS 5-fold cross-validation is used to select these recipes. The tune data is used to find optimal values for λ_u and λ_v. For NMF eleven sets of 20 recipes are selected randomly. Ten sets form the tune data (one at a time) and one set the test data. This is done 100 times. The number of tune/test recipes is kept low to minimize their influence in the decomposition of the matrix. The tune data is used to find the optimal value of k.

One random ingredient of each of the tune and test recipes is eliminated. For each recipe, the remaining ingredients are fed to the model. This allows to test the ability of the model to retrieve an eliminated ingredient. The model returns a list of ingredients, which is ordered so the first ingredient makes the best combination with the given ingredients. The rank of the eliminated ingredient in this ordered list is used to select the optimal values during tuning (minimization of the rank) and to evaluate the performance of both methods during testing. When using NMF the eliminated ingredient can be found in the top ten of best fitting ingredients in 43.6% of the test recipes. For two-step RLS this is true for 57.5% of the test recipes. This can be seen in Figure 2.

Testing whether or not a model can retrieve an eliminated ingredient is one way to test the performance of a model to complete a recipe. However, it is also important to look at the other ingredients, apart from the eliminated ingre-

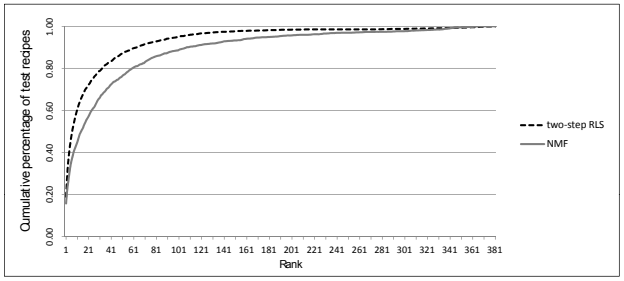

Figure 2. Cumulative distribution of the rank of the eliminated ingredient in the ordered list of best fitting ingredients for NMF and two-step RLS. The cumulative distribution represents the percentage of test recipes for which the eliminated ingredient has a rank in the ordered list smaller or equal to a certain rank.

Table 1. Top five of best fitting ingredients to add to the given set of ingredients to complete the recipe using two-step RLS.

egg, cocoa, cream	chicken, rice, cream	tomato, beef, wheat
butter	brown rice	raw beef
wheat	onion	onion
vanilla	chicken broth	egg
milk	butter	yeast
cane molasses	milk	garlic

dient, that are on top of the list of suggested ingredients. Therefore, three sets of three ingredients are selected and given to the two-step RLS model. The selected ingredient sets and the top five of suggested ingredients can be found in Table 1. The suggested ingredients are quite acceptable, except for the first ingredient of the second and third set. These unwanted ingredients are present due to a high number of shared flavour components. One possibility to eliminate the undesired ingredient is to select the category to which the suggested ingredients should belong. An example is given in Table 2, where ingredients are suggested to add to a recipe already containing cocoa, coconut and vanilla. These ingredients can, for instance, be combined into a chocolate candy or an alcoholic cocktail.

When examining the two-step RLS model in more detail, it is clear that the presence of some ingredients will not influence the list of suggested ingredients. Examples of such ingredients are angelica, beech, geranium, holy basil, etc. These ingredients are rare and only used in a very small number of recipes. A consequence is that these ingredients will not be suggested to add to a recipe. Another conclusion that can be made is that the presence of a certain ingredient can prevent the presence of another ingredient in the list of suggested ingredients. A final conclusion is that ingredients found in a same cuisine have a higher chance to be combined, while completing a recipe than ingredients coming from different cuisines.

Table 2. Adding ingredients from a certain category to complete a recipe containing cocoa, coconut and vanilla by means of two-step RLS.

nuts, seeds and pulses	alcoholic beverages	spices
walnut	rum	coriander
pecan	wine	turmeric
almond	tequila	cumin

4. Discussion

NMF can be used to retrieve an eliminated ingredient, however, it has two disadvantages. A first disadvantage is that the method can only use a single matrix. Taking into account the food pairing theory, it could be interesting to add information on flavour components to the model as well. A second disadvantage is the difficulty to make predictions on a new recipe. The new recipe needs to be added to the matrix after which the whole matrix needs to be factorized and recombined once again. This is a quite computationally intensive process. A way out could be to determine the values in matrix H. This would allow to determine values for a new recipe, without factorizing the matrix.

Two-step RLS performs two regressions and allows to take into consideration two data sets. However, in our model, adding flavour data resulted in suggesting ingredients that were too alike to those already present in the recipe, for instance beef and raw beef. This problem was solved when the category, to which the suggested ingredients should belong, was given as well. Also a stronger validation of the model could eliminate the unwanted suggestions. Besides allowing the use of a second data set, two-step RLS also makes it much easier to make predictions on new recipes.

With better data, the performance of the model could improve, leading to better results, and we could get a better understanding of recipe completion. With better data, the application possibilities would also expand. Data-driven methods could then not only be applied to complete a recipe, but also in attempts to personalize recipes.

5. Conclusion

NMF is capable of retrieving an eliminated ingredient in a recipe, which can be seen as recipe completion. Just as NMF, two-step RLS is capable of retrieving an ingredient that was removed from a recipe as well. The results of two-step RLS are even better, therefore this model was used to complete three ingredient sets, each containing three ingredients, into a recipe. For each set the top five of best fitting ingredients was studied. The results of this experiment are very promising in terms of usability of data-driven methods to complete recipes.

Acknowledgments

MDC, MS, BDB and WW acknowledge the support of Ghent University. This work was carried out using the Stevin Supercomputer Infrastructure at Ghent University, funded by Ghent University, the Hercules Foundation and the Flemish Government - department EWI.

References

Ahn, Y.-Y., Ahnert, S. E., Bagrow, J. P., and Barabasi, A. L. Flavor network and the principles of food pairing. *Scientific Reports*, 1(196), 2011. doi: 10.1038/srep00196.

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999. doi: 10.1038/44565.

Pahikkala, T., Stock, M., Airola, A., Aittokallio, T., De Baets, B., and Waegeman, W. A two-step learning approach for solving full and almost full cold start problems in dyadic prediction. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2014.