Recipe Based Prediction

Veer Tilak Singh
Indiana University
107 S Indiana Ave,
Bloomington, USA
vtsingh@indiana.edu

Prerna Rustagi
Indiana University
107 S Indiana Ave,
Bloomington, USA
rustagip@umail.iu.edu

Abstract

In this task, we are trying to answer the question "Can we predict user rating of recipes based on ingredients present in them? " We have divided recipes in categories because comparing a salad recipe with soup or desert increases the probability of getting an inaccurate result. We make use of recipe category in two ways, in one way we are using category as features for machine learner, a multiclass support vector machine to examine. In another case we are creating different learning set based on these recipe categories. Later, in the paper we are comparing the results of two different input and output sets we created based on categories. We find that comparing a salad recipe with salad recipe provide more accuracy in predicting user rating in comparison to comparing it with all different kind of recipes with category as part of feature set. Our results show that if you compare recipes with in same category, $SVM^{multiclass}$ gives better accuracy on prediction of ratings.

1 Introduction

Popularity of sites like quora.com, answers.yahoo.com shows that, people go to internet, to get answer to all types of questions. It is no surprise that people go to internet to learn about new recipes. Hence various websites are providing recipes over the internet and all this data is available at no cost to users. This study is important because it provides us with the understanding that presence of particular ingredients makes a recipe get better rating. Better rating means more people like it. It will allow website to invent better recipes based on rating of previous recipes and ingredients present in them.

Among different recipe websites, we have used data from http://www.epicurious.com/ because Epicurious has over 100000 recipes listed in various *categories.* We are considering categories as soups, salad, sandwich and dessert etc.

In this paper, we predict recipe rating for individual recipes, given its ingredients. We are using two different methods for prediction of recipe rating based on the category recipe belongs to. Both the methods are as explained below:

1. In first method, we use category of recipe as a feature for SVM multiclass along with other ingredients.
2. In second method, we use recipe category as a classifier to create separate train dataset for each category.

Hence, this research paper concentrates on determining if it is better to compare recipe within same category for prediction or not and how an ingredient affects user opinion about a recipe.

2 Related Work

With our research we are trying to find out that whether presence of an ingredient makes a recipe good or bad. There has been lots of research on recipe recommendation based on ingredients. Algorithm has been developed to suggest recipe based on overlapping ingredients based on their search [1] and find recipe based on ingredients by including and excluding them in the recipe [2]. Algorithms have been proposed to suggest replaceable ingredients in a recipe to satisfy user requirements for example to reduce the calorie count or increase fiber amount in the dish [3].

3 How To Predict User Ratings

In this section, we describe the data sets created and then the research questions we are investigating in the paper.

3.1 Research questions

Can we predict the user rating based for a recipe based on its ingredients? Second question we are trying to answer here is that, is our rating prediction algorithm works better if use category of each recipe as a feature-value pair or is it more accurate if dataset is separated based on recipe category for machine learning tool and tested based on recipe category.

3.2 Data Set

Dataset we used for our research problem was corpus from popular recipe website http://www.epicurious.com/. This corpus contained about 28,392 files which had ingredients for different recipes and the measures for each. To create feature set of ingredient from these recipes, we made some key decisions and which are listed as following:

- Some ingredients in the recipes did not have any measures for them. We are using ingredients as feature and measures as value of the feature so we decided not to consider such ingredients which do not have any measure.
- We extracted the recipe category from the recipe metadata files which has type field defined. Recipes which did not have type field in their metadata file we extracted the category from dietary consideration field. Metadata files which did not have both dietary considerations and type field, we excluded such recipes.
- Some of the recipes had type field description as ‘Kid-friendly’, ‘Quick and Easy’ in their metadata file, for such files we put them in “undefined” category.
- There were different kinds of measures used for ingredients like tablespoons, cups, sticks etc. We have normalized all the measurements to tablespoons.
- The only forks considered were 1 fork, 2 forks, 3 forks and 4 forks. Half forks were rounded off to the lower number. For example 3.5 forks is considered as 3 forks.
- We have considered the quantities (measures) of ingredients as feature values.
- We are considering all the ingredients which are present in a recipe if they have measure defined.
- All recipes with 0 rating were excluded from calculation.

4 Experimental Setup and Evaluation

For classification of data we used SVM$^{\text{Multiclass}}$ based on [1]. The experiments performed in this paper were accomplished using default settings of SVM multiclass tool. We have used linear model for these experiments. For tagging of the ingredients we have used Stanford Part-Of-Speech Tagger.[5]

5 Approach

We first extracted the recipe types from metadata files and appended these category values to the file name. For example: 0_Bacon-Cheddar-Burgers-with-Caramelized-Onions-51112220.txt was converted to 0_Bacon-Cheddar-Burgers-with-Caramelized-Onions-51112220(Sandwich).txt.

We used Stanford POS tagger to tag the ingredients in the modified files. As Stanford tagger is trained on Wall Street Journal section of the

Penn Treebank, it is possible that it would have tagged some ingredients wrong.
From the POS tagged files, we extract “NN”, “NNS” and “JJ” tags to get the ingredients of a recipe. We extracted ingredient measurement values and calculated an average of measure of ingredient if it was not defined clearly. For example consider this line in the recipe file “butter 2 to 3 sticks”, the ingredient feature/value pair would then be “butter 2.5”.
We assume that, it is important to not distinguish between similar features such as “lemon juice” and “freshly squeezed lemon juice” or just “lemons” since ingredients in their original description are essentially different. For example, a recipe which uses “lemon juice” may not taste the same if just “lemons” are used. So we have used “lemon juice” and “freshly squeezed lemon juice” as separate features.
Recipes files had different measurement units for ingredient measures so we have normalized all the measures of the ingredients to tablespoon measure. Conversion of these measures is done using following table:

1 cup	16 ts
1 pound	32 ts
2 sticks	16 ts
1 ounce	2 ts

We did not place any restriction on the number of ingredients present in a recipe file.
Out of total 28,392 recipe files, 23084 recipe files were considered because their metadata had either recipe “type” or “dietary considerations” defined.
Number of files, which were not considered were the ones which had no “type” or “dietary considerations” defined in the metadata = 5308 files.
In the first input dataset where we used category as feature with its value as 1 for SVM multiclass, we used all 23084 files, in which 18496 files were used for train dataset and rest 2291 files for test dataset.
In the second input data set, we created separate train and test dataset based on their recipe category. We got 48 different categories; we did not run experiments on some of these categories as there were not enough data to train and test SVM.

6 Results

Result for category as a feature (cumulative all forks):

Precision	Recall	F-Score
.597	.567	.581

Result for category as a super class (cumulative, all forks):

Category	Precision	Recall	F-Score
Bar Cookie	0.647	0.647	0.647
Bread	0.483	0.483	0.483
Cake	0.496	0.496	0.496
Candy	0.266	0.266	0.266
Cookie	0.671	0.671	0.671
Dressing	0.357	0.357	0.357
Egg Dish	0.448	0.448	0.448
Frozen Dessert	0.396	0.396	0.396
Grating	0.608	0.608	0.608
Healthy	0.473	0.473	0.473
High Fiber	0.304	0.304	0.304
Noodle Dish	0.542	0.542	0.542
Salad	0.534	0.534	0.534
Sandwich	0.411	0.411	0.411
Sauce	0.396	0.396	0.396
Spread	0.444	0.444	0.444
Stew	0.556	0.556	0.556
tart	0.474	0.474	0.474
undefined	0.455	0.455	0.455
Vegetarian	0.555	0.555	0.555

The rating prediction for the categories such as “Bar Cookie”, “Cookie” and “Grating” have higher accuracy since the number of training and testing files were more in those categories. The other categories suffer a little since the training and testing records weren’t many after splitting.

7 Conclusion and Future Work

In this work we extracted features from ingredients and metadata. From the above results we conclude that it is more accurate to predict the ratings if recipes in the same categories were considered while building the training data set.
As part of the future work we would like to distinguish between similar ingredients and see how they help in better predicting recipe ratings.
When the categories were split, some recipe categories did not have enough training data due to which the accuracy of the predictions reduced. Therefore, it would lend more clarity to work with a larger dataset such that every category has enough training and test data. This work can be extended to working with cooking instructions too and recipes in the same category could be used to predict recipes from the same category and would be interesting to see how far that would get us.

References

[1] Freyne, J., and Berkovsky, S. Intelligent food planning: personalized recipe recommendation. In IUI, ACM (2010), 321–324.

[2] Zhang, Q., Hu, R., Mac Namee, B., and Delany, S. Back to the future: Knowledge light case base cookery. In Proc. of The 9th European Conference on Case-Based Reasoning Workshop (2008), 15.

[3] Shidochi, Y., Takahashi, T., Ide, I., and Murase, H. Finding replaceable materials in cooking recipe texts considering characteristic cooking actions. In Proc. Of the ACM multimedia 2009 workshop on Multimedia for cooking and eating activities, ACM (2009), 9–14.

[4] K. Crammer and Y. Singer. On the Algorithmic Implementation of Multi-class SVMs, JMLR, 2001.

[5] http://nlp.stanford.edu/downloads/tagger.shtml