What makes a good muffin?

Ivan Ivanov

CS229 Final Project

Introduction

Today most cooking projects start off by consulting the Internet for recipes. A quick search for "chocolate chip muffins" returns a multitude of different recipes, and typically, we would look though the top rated ones and try to decide which one "looks best". We would go down the ingredients list, make a few substitutions, depending on what's left in the fridge, and maybe scale down the recipe, since we don't have a whole dinner party to feed… But is it really OK to substitute sour cream for yogurt? What if I'd like to use soy milk instead of whole milk? And how do you split three eggs in half?

This project developed a learning algorithm which can predict the success of a muffin recipe based on the quantity of each ingredient used. The algorithm was trained on data from muffin recipes collected from www.allrecipes.com. The input to the algorithm is a list of ingredients with their corresponding amount and the output of the algorithm is a numerical score measuring recipe success. In order to make a recommendation for a good muffin recipe, this algorithm can be used to optimize ingredient quantities by maximizing the scoring function.

Related Work

Predicting the success of an object (recipe, book, song, etc.) based on its constituents (ingredients, words, or sound frequency) can be a difficult problem and various approaches to it are found in the literature. Cortez et al. recently reported on predicting wine preferences based on the chemical characteristics of the wine [1]. In the analysis, the authors used multiple regression, neural network methods and support vector machines (SVM) as learning models and concluded that SVM was the most reliable predictor for that data set. In another study, Teng et al. found that recipe ratings can be predicted based on features derived from combinations of ingredient networks and nutrition information [2]. They also point to the fact that user reviews can be a good resource of information on possible ingredient substitutions, or the appropriate range of quantity of some ingredients. A similar network analysis of recipe ingredients was performed by Ahn et al [3]. In this study, the authors find that Western cuisines often use ingredients that share a flavor profile, while East Asian cuisines do not. Information on user preferences can be valuable and has been exploited in various product-recommendation algorithms [4-6].

Dataset and Features

Data on 540 muffin recipes was collected from http://allrecipes.com/recipes/350/bread/quick-bread/muffins/. This is an example of the extracted features:

1. Name: Chocolate Chip Muffins
2. URL: http://allrecipes.com/recipe/7906/...
3. Recipe ID: 7906
4. Rating: 4.029586
5. Review count: 71
6. Made-it count: 40
7. Servings: 12

Information on each ingredient was processed in order to derive the following features:

1. Name: flour
2. Ingredient ID: 1684
3. Amount: 2
4. Unit: cup
5. Modifiers: all-purpose

A total of 454 unique ingredients were present in the collected set of recipes. Ingredient names were stemmed using the Porter stemming algorithm [7] in order to remove suffixes (e.g. "egg" and "eggs") and facilitate downstream processing. Based on this data, the amount per serving in ounces was calculated for each ingredient. Furthermore, similar ingredients were grouped in categories – for example, "all-purpose flour" and "whole-wheat flour" (which have distinct ingredient IDs) were grouped under "flour". This reduced the number of unique ingredients to 180. In order to prevent overfitting of the learning model, only ingredients which appear in more than 10 recipes were considered. Thus the final size of the design matrix for this project is $\boldsymbol{X} \in \mathbb{R}^{540\times 50}$.

Methods

The output variable of the learning algorithm is a recipe success score, calculated by multiplying the average user rating by the confidence metric $c(n\_reviews)$, which depends on the total number of user reviews for the given recipe.

$$y = rating * c(n\_reviews)$$

$$c(n\_reviews) = 1 - \exp(\alpha * n\_reviews)$$

where $\alpha = 0.05$. Thus the score of recipes with less than 20-50 reviews is decreased exponentially.

In order to predict the recipe success score based on the amount of ingredients used, linear regression, logistic regression, and support vector machine classification were employed. Least squares linear regression derives the model parameters by minimizing the square error between the data and the model prediction:

$$h(x) = \theta^T x$$

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

An analytical solution of this problem exists in the form of the normal equations:

$$X^T X\theta = X^T y$$

The goodness of fit of the regression models was judges using the error metric:

$$error = 1 - R^2 = \sum\nolimits_i \left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right)^2 \Big/ \sum\nolimits_i \left(y^{(i)} - \bar{y}\right)^2$$

Logistic regression derives the model parameters by maximizing the log-likelihood of the data:

$$h(x) = 1/(1 + \exp(-\theta^T x))$$

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} \log\left(h\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h\left(x^{(i)}\right)\right)$$

No analytical solution to this optimization problem exists, and the model parameters are obtained using algorithms such as gradient ascent or Newton's method.

SVM classification is achieved by find the optimal margin classifier:

$$h(x) = g(w^T x + b), \quad g(z) = \begin{cases} 1 & if\ z \geq 0 \\ -1 & otherwise \end{cases}$$

$$\min_{\gamma,w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \xi_i$$
$$s.t.\ \ y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i,\ \ i = 1, \ldots, m$$
$$\xi_i \geq 0, i = 1, \ldots, m$$

The optimization is typically accomplished by solving the Lagrange dual problem. This algorithm also allows for mapping the data into higher-dimensional space using kernels. In this project, the Gaussian kernel was used: $K(x, z) = \exp\left(-\gamma||x - z||^2\right)$.

Success of the classification algorithms was judged by the percent correctly classified examples. All learning algorithms were implemented using the *scikit-learn* library in Python [8]. Models were trained on randomly chosen 80% of the data and tested on the remaining 20% of the data.

Results and Discussion

As an initial attempt at predicting muffin recipes, only a subset of all recipes was considered. A search for banana muffins returned 62 recipes, which contain 15 features, as defined above. Classification was performed on two classes, "good recipes" and "bad recipes". Good recipes are defined as recipes with score greater than 3.

Logistic regression predicted the outcome of banana muffins with moderate success. The model achieved greater than 60% accurate classification, as determined by hold-out cross validation (Figure 1).

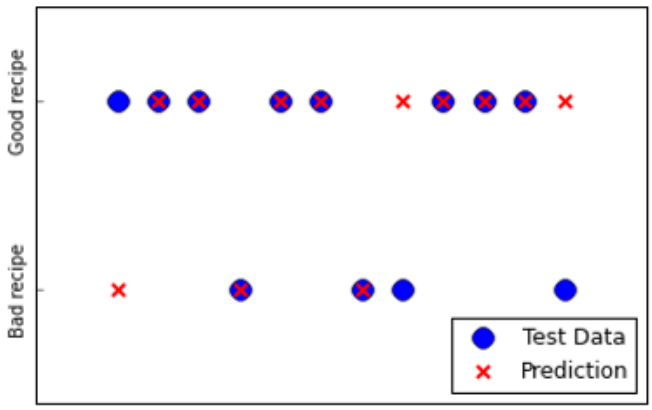

Figure 1. Example data on prediction of the success of banana muffin recipes using logistic regression. The model achieves greater than 60% accuracy.

The logistic function, however, is not convex, and this will pose a difficulty in the second stage of the project, which aims to optimize the ingredients of a recipe by maximizing the scoring function. In order to facilitate the optimization problem, a scoring function of lower complexity was considered.

Least-squares linear regression was used on this data set. Score predictions were thresholded to the interval of $[0, 5]$. This model provided a reasonable measure of the success of banana muffin recipes (Figure 2).

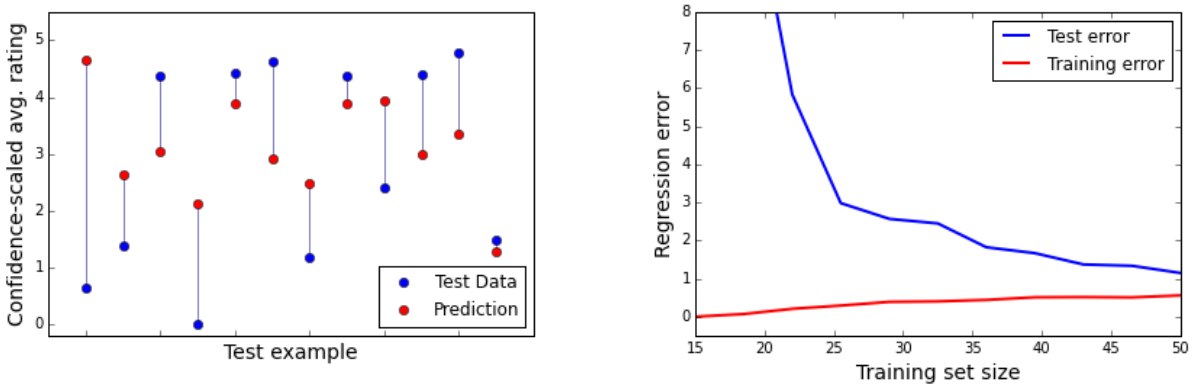

Figure 2. Prediction of the success of banana muffins using least-squares linear regression. Left: example data and model prediction points. Right: Learning curve for this model

Based on this model, it was determined that the top three ingredients which contribute most to the success of a banana muffin are butter, bananas, and sugar. The bottom three ingredients, which negatively scale with recipe success, are vanilla, salt, and cinnamon.

The linear regression model, however, did not performed well when trained on the entire data set. The model suffered from problems of high bias and high variance (Figure 3, Left). This is partly explained in a plot of the principal components of the data (Figure 3, Right), which does not show a clear trend. Constraining the L1-norm of the parameters (Lasso regression) did not improve the model further.

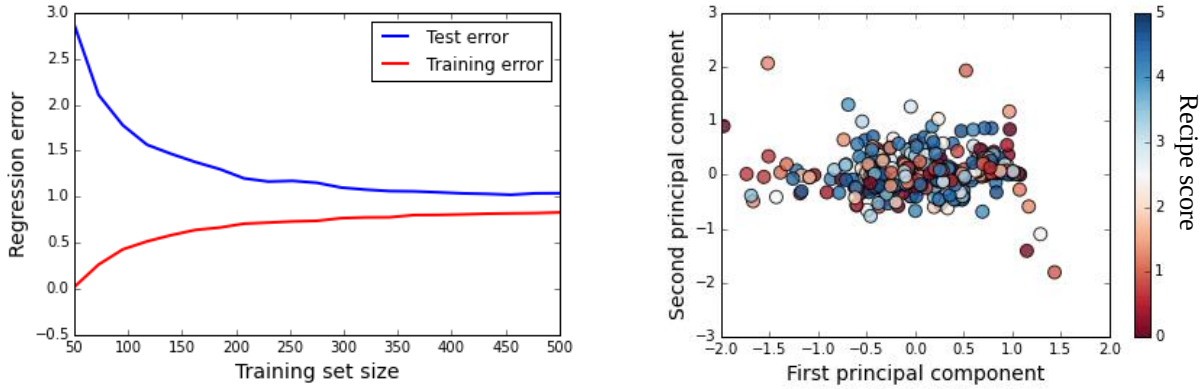

Figure 3. Predicting the success of all muffin recipes using linear regression. Left: the learning curve for this model suggests that it suffers from high bias and high variance problems. Right: PCA analysis of the data does not show a clear dependence of the recipe score on the first two principal components.

The success of any muffin recipe was also not well predicted by binary SVM classification with a Gaussian kernel (Table 1). The parameters C and γ of the model were optimized using hold-out cross validation. The model, however, displayed a tendency of classifying “bad” recipes as “good”, i.e. it has low specificity. The model has accuracy of 0.56, which is only slightly higher than the null error rate,

equal to 0.41. SVM classification of the data into six categories (0-star through 5-star recipes) also performed poorly.

Table 1. Confusion table of binary SVM classification on full data set

N = 108		Predicted:	
		Bad	Good
Actual:	Bad	22	42
	Good	6	38

We speculate that the difficulty in predicting the success of muffins recipes may result from the way user ratings are assigned. Users may be biased towards providing a rating which conforms more to the average rating of the recipe, rather than expressing their objective opinion on it. This is corroborated by the fact that the average rating of recipes in the data set is relatively high at 4.3 stars. Furthermore, users often exhibit flocking behavior and would tend to try recipes that already have high rating and a large number of reviews. In this way, there may be good recipes in the data set, which have not received a lot of user reviews, and thus get a low score in this algorithm. This problem may be addressed by expanding the data set. Lastly, it is likely that the success of a muffin recipe is not only determined by the quantity of the used ingredients, but may also be affected by other factors not considered in this project.

Conclusions and Future Work

This project developed a learning algorithm which predicts the success of a muffin recipe based on the quantity of ingredients used in the recipe. It was found that the model performs well on a subclass of the data set (e.g. banana muffins, chocolate chip muffins, etc.), but does not generalize well to predictions on the entire data set.

Successful optimization of this algorithm will allow it to be used to identify universal relationships (such as ratio of dry ingredients to wet ingredients which results in moist muffins, or amount of leavening agents to flour which makes the muffins raise nicely) and also suggest the optimal recipe for a specific subclass (e.g. best blueberry muffins, best cranberry muffins, etc.). The algorithm will also be able to suggest scaling relationships (e.g. to make 20 muffins, should I use 2 or 3 eggs, when the correct scaling calls for 2.7 eggs?; should I maybe use 2 eggs and increase the amount of butter a little?) and adjust the recipe based on desired substitutions (e.g. should I decrease the amount of sugar, if I want to use vanilla soy milk instead of 2% milk?).

This project focused on making predictions for muffin recipes, but the software developed here can be easily extended to making recommendations for other dishes, and in general, finding the optimal combination of a set of features, with appropriate scaling, and the ability to include optional features, if desired.

References

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems,* vol. 47, pp. 547-553, 11// 2009.

[2] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, "Recipe recommendation using ingredient networks," pp. 298-307.

[3] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, "Flavor network and the principles of food pairing," *Scientific Reports,* vol. 1, p. 196, 12/15/online 2011.

[4] J. Freyne, S. Berkovsky, and G. Smith, "Rating Bias and Preference Acquisition," *ACM Transactions on Interactive Intelligent Systems (TiiS),* vol. 3, p. 19, 2013.

[5] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," pp. 2643-2651.

[6] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E,* vol. 76, p. 046115, 10/25/ 2007.

[7] M. F. Porter, "An algorithm for suffix strippingnull," *Program,* vol. 14, pp. 130-137, 1980/03/01 1980.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, *et al.*, "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.