Cuisine Classification from Ingredients

Boqi Li, Mingyu Wang

Abstract

In this report, the team aimed to classify 20 types of cuisines by analyzing the ingredient lists. The team approached this problem by building several feature sets and trained them with different classification algorithms. The result shows that the logistic regression is able to achieve the highest performance with an accuracy of 78.4%. The error distribution of each class is analyzed for further investigation.

I. Introduction

Yummly has provided a large dataset of recipes in which each entry has the type of cuisine and the list of its ingredients. The goal of the project is to categorize a dish' s cuisine type (Indian, Italian, Chinese, etc.), by analyzing its ingredient list. Since there are 20 classes in the cuisine set, the team will use several multiclass classification techniques including OVA multiclass Naive Bayes, OVA R2-regularized Logistic Regression, OVA multiclass SVM, OVA multiclass SVM with crammer method and K-Nearest-Neighbor and compare the performance among these methods. The input of our algorithm is a list of ingredients of some recipe, and the output is the predicted cuisine. The team will also modify the features to improve performance. This project has an intriguing application. Image when people search on Yummly for certain type of cuisine or browsing related recipes, this algorithm can help Yummly decide which recipe to display or recommend recipes that people may be interested in to improve user experience.

II. Related Work

After a literature search, the team found several ways to approach multi-class classification problem. One well studied method is called the Error Correcting Output Coding method, which divides the multi-class classification problem into a series of binary classification problems, which is suggested by Rennie and Rifkin[1]. The advantage of this method is that, by reducing the problems to binary classification, many well known algorithm can be applied, such as Naive Bayes and SVM. The problem is that since this method trains the classifiers independently, it doesn't take the correlation between classes into account. Another newer method that considers the correlation is introduced by Crammer and Singer, which is a modified multi-class SVM.[2] Apart from these two methods, Trudgian and Yang suggests using K nearest Neighbor to approach the classification problem. The advantage of K nearest Neighbor is that it can operate quickly without sacrificing accuracy greatly.[3]

There are rigorous researches in text classification, in [4], a variety of feature selection and learning schemes are discussed. To reduce dimension of feature space, document frequency thresholding, Information gain, Mutual Information and Term Strength are compared. In our paper, we applied tfidf and mutual information on our dataset. In [5], tfidf is discussed in detail. While this method is borrowed from information retrieval, it doesn't utilize the privilege of labeled training set in supervised learning. A new term weighting schemes is proposed and compared with other schemes. In this paper, the group adopted the classical tfidf term for simplicity.

III. Dataset and features

Our dataset was obtained from Kaggle competition "What's Cooking" (https://www.kaggle.com/c/whats-cooking/data). 39774 examples from 20 cuisines are available in total. The training set has 6714 unique ingredients which are considered as features in our dataset. Uniqueness is in the sense that the strings are different from each other. One recipe is represented by a feature vector. In constructing feature vectors, we considered different approaches. Notations in this paper follow the convention of our class. Let $\mathcal{S} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), ..., (x^{(m)}, y^{(m)})\}$ be the training set. And each $x^{(i)}$ is a subset of the vocabulary $\mathcal{V}$, which means that it contains some ingredients combination. $x_j^{(i)} = 1$ indicates feature j occurs in example i and 0 otherwise, $j = 1, 2, 3, \dots, n$. $y^{(i)} \in \{1, 2, 3, ..., 20\}$, which correspond to the 20 cuisine types.

- Binary representation: if the ith ingredient appears in the example, then the ith term in the feature vector is set to one, and zero otherwise. In the recipes context, a ingredient cannot appear more than once. So this representation can be regarded as a frequency of the ith feature.
- tf-idf frequency. $tf(x_i^j)$ is the term frequency of ingredient i in recipe j. $tf = 1$ if recipe contains this ingredient and 0 otherwise. $idf(x_i, S)$ is inverse document frequency to measure how common a ingredient is in our dataset. $idf(x_i) = log\frac{|S|}{|x \in S : x_i = 1|}$ where m is the number of examples in the training set and $|x \in S : x_i = 1|$ is the number of recipes which certain ingredient appears. Then, tf-idf is calculated as $tfidf(x_i^j, S) = tf(x_i^j) \times idf(x_i, S)$

Having observed that our dataset is highly biased(as in Figure 1). The group also assigned different weight to achieve a uniform cuisine distribution. However, it doesn't show improvements in the prediction result.

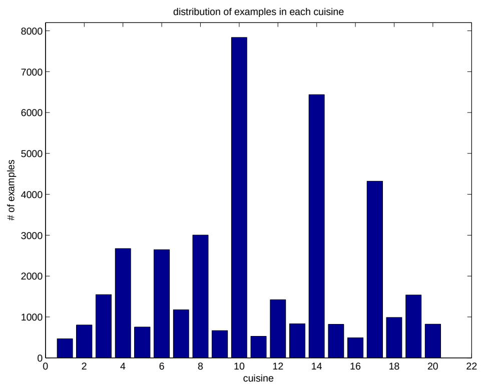

Fig. 1. Distribution of cuisine examples in training set is highly biased. Italian, Mexican and Southern_US cuisines have the most examples while others have much less examples.

Then, before selecting features, the group removes any digits, measurements, punctuations and content inside parenthesis. The brands in the dataset are also cleared and all strings are converted to lowercase. In reducing the number of features, several alternatives are studied.

- One major problem with the feature data is that it contains many low-content or synonyms of descriptive words, for example, {"low fat","reduced fat", "fat reduced"} mean the same thing and {"'natural', "fresh", "artificial"} are considered to be low-content and should be removed to combine ingredients. After these steps, strings like "1% low-fat chocolate milk" are converted into "chocolate milk". To stem different variants of a word, stemming algorithm is performed. In this paper, Porter Stemming algorithm is used to remove plural form of words. These steps let us reduce the number of features to 5353.
- An alternative way is Mutual Information of features. Plot of mutual information is shown in Figure 2. Only the 4000 ingredients with the largest mutual information are taken into account in the following results.
- The group also considered selecting the features according to the number of occurrence. Only ingredients that appeared more than 5 times are considered, which gives us 3063 features. This can help us exclude ingredients with typos, brands or redundant descriptive information. To study the influence of number of features on our prediction model, different number are chosen and studied.

Furthermore, 30000 examples are used as training set and 9000 examples are used as testing set to evaluate the classification models.

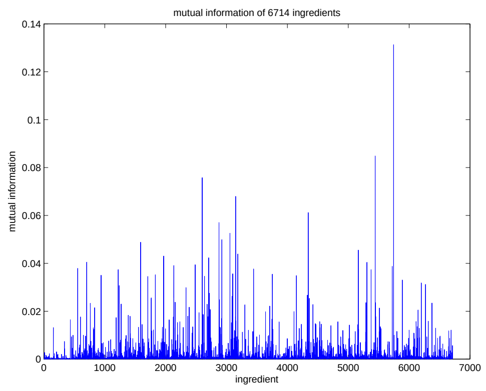

Fig. 2. plot of mutual information of 6714 ingredients

IV. Methods

Multi-class classification involves labeling among m classes with $n > 2$. And in this project $n = 20$. The team then applied Error Correcting Output Coding introduced by Dietterich and Bakiri. As is mentioned in previous section, the Error Correcting Output Coding method links multi-class classification with binary classification. It allows approaching multi-classification problem by training a series of binary classifier. In this project, the team choose the Code Matrix $R \in \{1, -1\}^{20} * \{1, -1\}^{20}$to be One vs. All matrix (OVA) with dimension of 20*20. The matrix has 1 on its diagonal, and -1 on all other places. The columns of R corresponds to 20 binary classifiers $(f_1 f_2 ... f_{20})$. And the rows of R corresponds to the 20 classes. For each classifier, there will be only one class labeled as 1, with all the rest labeled as -1. For each training instance $(x^{(i)}, y^{(i)})$, the label trained with class f_j is $R[y^{(i)}, j]$. Each classifier will be trained with all the training instances. When predicting the label of a new instance x, each classifier will predict on x, and form a row vector $r_x = \{f_1(x), f_2(x), f_3(x), f_4(x), ...f_{20}(x)\}$. The row vector will be compared with every row of matrix R $(r_1 r_2 ... r_{20})$by computing the dot product between r_x and $r_i, i = 1, 2..., 20$.

$$s_i = \sum_{k=1}^{20} r_x[k] * r_i[k]$$

If the k-th elements of r_x and r_i have the same sign, a positive number will be added into the summation, otherwise a negative number will be add. Therefore the row r_i that is closest to r_x will have the largest dot product.
When comparing between r_x and matrix R, if $f_k(x) = 1$ or -1, then only the signs are considered and the magnitude information $\theta^T x$ is lost, which means the confidence of the label predicted is not taken into consideration. Therefore, the team managed the output of each classifier to be $\theta^T x$,for the case of SVM, this becomes $\omega^T x + b$, and for Naive Bayes, $P(y = 1|x) - P(y = 0|x)$ is used. And the

hinge loss function is applied to predict the class. Finally, the predicted label of instance x is:

$$H(x) = \underset{c\in\{1,2,..,20\}}{\operatorname{argmin}} \sum_{j=1}^{20} g(r_x[j] * R[c,j])$$

where $g(s) = |1-s|_+$. Binary classification algorithms including Naive Bayes, logistic regression and SVM are applied based on Error Correcting Output Coding. The description of each algorithm is described in the remaining part of this section.

A. Naive Bayes Classification

Naive Bayes is a binary classification algorithm, which is good for text classification purpose. Let function f be the classifier that can achieve the labeling of 2 labels. And with a new example x, the predicted label is:

$$f(x) = P(y=1|x) - P(y=0|x),$$

where

$$\begin{aligned} & P(y=1|x) \\ = & \frac{P(x|y=1)P(y=1)}{P(x)} \\ = & \frac{\prod_i \phi(i|y=1)\phi(y=1)}{\prod_i \phi(i|y=1)\phi(y=1) + \prod_i \phi(i|y=0)\phi(y=0)}, \end{aligned}$$

and

$$\begin{aligned} & P(y=1|x) \\ = & \frac{P(x|y=0)P(y=0)}{P(x)} \\ = & \frac{\prod_i \phi(i|y=0)\phi(y=0)}{\prod_i \phi(i|y=1)\phi(y=1) + \prod_i \phi(i|y=0)\phi(y=0)}, \end{aligned}$$

And after training every parameter of Naive Bayes with the training set of size m. The parameters are shown as:

$$\begin{aligned} \phi(j|y=1) &= \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}} \\ \phi(j|y=0) &= \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}} \\ \phi(y=1) &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m} \end{aligned}$$

Laplace smoothing is used and $|\mathcal{V}|$ is the size of the vocabulary.

B. Logistic Regression

Logistic Regression is a robust classifier that does well with a larger training set, which makes weaker model assumption. It generally performs well even if the model distribution is non-Gaussian. The hypothesis function still uses sigmoid function but in the end the logistic regression classifier will return the value of $\theta^T x$ instead of $g(\theta^T x)$, in order to be fetched to the hinge loss function. The hypothesis function is:

$$h_\theta = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

And the stochastic gradient ascent rule is:

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x^{(i)}$$

C. Support Vector Machine

Support Vector Machine is binary classifier that finds the optimal margin between two classes of data. The hypothesis function is $h = g(\omega^T x + b)$, where g=1 when $\omega^T x + b \geq 0$ and g=-1 when $\omega^T x + b \leq 0$. Still the classifier will return the value of $\omega^T x + b$, which will be used in the hinge loss function. In this project, the regulated Support Vector Machine is applied and the optimization problem becomes:

$$min_{\gamma,\omega,b}\frac{1}{2}||\omega||^2 + C\sum_{i=1}^{m}\xi_i$$

s.t.

$$y^{(i)}(\omega^T x^{(i)} + b) \geq 1 - \xi_i, i = 1, ...m$$
$$\xi_i \geq 0, i = 1, ...m$$

According to Yang and Liu [6], the linear kernel works better than non-linear kernel on text classification application. Therefore, linear kernel is used on SVM.

D. Multi-class Support Vector Machine by Crammer and Singer

The next model is a multi-class support vector machine classifier based on Crammer and Singer. The difference between their SVM method and the OVA SVM method is that, in OVA SVM the correlation between each class is not considered, because the classification is divided into training multiple independent binary classification problem. In the SVM by Crammer and Singer, this correlation is taken into account. The multi-class SVM is trained via the following optimization problem:

$$max_\tau \mathcal{Q}(\tau) = -\frac{1}{2}\sum_{i,j} K(\bar{x}_i, \bar{x}_j)(\bar{\tau}_i \cdot \bar{\tau}_j) + \beta \sum_i \bar{\tau}_i \cdot \bar{1}_{yi}$$

Subject to:

$$\forall i \bar{\tau}_i \leq \bar{1}_{yi}$$
$$\bar{\tau}_i \cdot \bar{1} = 0,$$

and the function H is:

$$H(\bar{x}) = \underset{r}{\operatorname{argmax}} \left[\sum_i \tau_{i,r} K(\bar{x}, \bar{x}_i)\right].$$

The team used the liblinear module with linearoption '-s 4', which corresponds to the multiclass SVM classifier.

E. K nearest neighbor

The last algorithm used is the K nearest neighbor algorithm. The K nearest neighbor compares the new instance as query point with all other instances in the training set and selects the k nearest instances to the query point. Using OVA coding Matrix, there will be 20 k nearest neighbor classifiers, each classifies one binary classification problem. And

$$L_1 = \sum_j \frac{1}{D_{1,j}}$$

$$L_{-1} = \sum_j \frac{1}{D_{-1,j}}$$

where $D_{1,j}$ is the distance between the jth neighbor of the query point that has label 1, and $D_{-1,j}$ is the distance between the jth neighbor of the query point that has label -1. The output is

$$L_1 - L_{-1}$$

, which will be used in the hinge loss function. Since every instance x in the training set is a set of a combination of the feature set, it is tested that the Jaccard distance is the desired distance metric for K nearest neighbor that outputs the best performance. The Jaccard distance is one minus the Jaccard coefficient, which is shown as:

$$J_{distance} = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

V. Result and discussion

Using the classification algorithm discussed in the previous section, the team is able to carried out results with 5 different feature sets on each algorithm. Every feature set is compared to the original feature set and the primary goal is the find the feature set and classifier that yields the highest test accuracy.

The first feature set contains all of the original 6714 features. This is set as a baseline for comparison with other feature sets created as described in the feature selection section.

The second feature set contains 3060 of original features selected with the highest occurrence in the whole training set. Every feature in this feature set appears at least 5 times in the training set.

The third feature set contains 4000 of the original features selected with the mutual information between the features and classes higher than 0.01.

The fourth feature set contains 3050 new features created by combining all the same key words, regardless of any descriptive words.

The last feature set contains 5353 new features created by first remove all the low-content descriptive words and then combining the same key words which is manually controlled so as not to be over-simplified.

For the two SVM classifiers, the regulation coefficient C is equal to 0.2. The team tested with several coefficient C ranging from 0.01 to 5. The result shows that with C=0.2, the SVM classifiers yields best accuracy. For K nearest neighbor, the number of nearest neighbor K is equal to 5. The team tested with several values of K ranging from 5 to 10, it appears that the resulting accuracy doesn't increase much with K increases from 5 to 10. However the computation time of the code increases a lot. Therefore K=5 is selected for KNN.

For every classification algorithm and feature set, 30000 examples in the dataset is selected to form the training set, and 9000 examples is selected to form the test set. This data allocation is pre-determined and the same for every classification algorithm and feature set combination, no randomization is utilized. The result is shown in Table I.

Table I shows that the best performance observed is from the logistic regression with the combination and reduction feature set. The logistic regression and the two SVM classifiers have much better performance than Naive Bayes and K nearest neighbor. Different feature sets have very limited influence on the result of logistic regression and the two SVM classifiers. The improvement compared to the original feature set is very low. The mutual information feature set increases the accuracy of the Naive Bayes classifier by 9%, which is the highest improvement among all feature sets in the Naive Bayes case. The combination and reduction feature set has the highest increase of accuracy to the K nearest neighbor, which is about 5%. The test error of Logistic regression

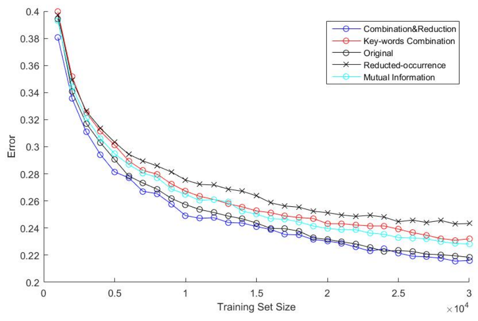

Fig. 3. Test Error for Logistic Regression with different feature sets

with increasing training set is shown in Figure 3. It can be seen that applying any of the feature selection other than combination and reduction doesn't improve the result compared to the original feature set. And the combination and reduction feature set only improves the performance slightly. With increasing training set size, the test error can still decrease.

The confusion matrix is shown in Table II. The element in row i column j shows the percentage at which cuisine i is classified as cuisine j. Blue cell boxes show the correct classification rates and red cell boxes show the error classification rates greater then 10%. Compared with the cuisine distribution in Figure 1. It can be seen that the classes with high mislabel error rate are those with fewer numbers of examples. It also shows that several classes are often mislabeled as class 6, 10 and 17, whereas class 10, 17 also have a high accurate rate, which means that those classes are similar to these classes.

TABLE I
Overall performance for each classification algorithms

	OVA NB	OVA SVM	OVA Logistic	SVM by Crammer	KNN
Original features	0.623	0.751	0.772	0.746	0.661
Reduced-occurrence	0.670	0.756	0.778	0.746	0.683
Mutual Information	0.713	0.757	0.771	0.743	0.697
Key Words Combination	0.653	0.750	0.768	0.740	0.690
Combination Reduction	0.622	0.765	0.784	0.754	0.714

TABLE II
Confusion Matrix (unit: %)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
46.9	0.0	3.5	0.0	0.0	3.5	0.0	0.9	0.9	6.2	0.9	0.0	0.0	20.4	0.0	0.0	12.4	3.5	0.9	0.0
0.0	43.3	0.5	0.5	0.5	14.4	0.5	2.6	4.1	9.3	0.0	1.0	0.5	1.0	0.5	1.0	19.6	0.0	0.0	0.5
0.3	0.0	71.3	0.0	0.0	3.9	0.0	0.0	0.0	6.7	0.3	0.0	0.0	1.7	0.0	0.0	14.5	1.4	0.0	0.0
0.0	0.2	0.5	85.1	1.4	0.0	0.2	0.8	0.2	1.3	0.0	3.0	2.2	0.5	0.0	0.0	0.9	0.2	2.7	0.9
2.4	0.0	0.0	13.3	51.8	7.2	0.6	0.6	0.0	3.6	0.0	1.2	0.0	0.6	0.0	0.0	9.6	1.8	4.2	3.0
0.2	1.3	1.1	0.2	0.6	62.3	0.5	0.6	0.5	18.9	0.3	0.3	0.0	0.8	0.8	0.8	8.7	1.7	0.2	0.2
0.4	0.7	0.4	0.4	0.0	3.2	67.0	1.8	0.0	20.8	0.0	0.0	0.0	0.0	1.4	0.7	2.2	0.7	0.4	0.0
0.1	0.0	0.1	0.3	0.1	0.4	0.6	92.2	0.0	1.0	0.3	0.4	0.0	1.3	1.3	0.0	0.7	0.3	0.7	0.0
0.0	10.3	0.0	1.3	0.0	8.4	0.6	0.6	43.9	7.1	0.0	0.0	0.0	1.3	0.6	2.6	22.6	0.6	0.0	0.0
0.1	0.2	0.1	0.1	0.1	3.7	1.0	0.3	0.1	89.5	0.1	0.1	0.0	0.9	0.1	0.2	2.8	0.5	0.1	0.1
0.9	0.0	0.9	1.7	1.7	2.6	0.0	4.3	0.0	6.1	60.9	0.0	0.0	7.8	0.0	0.9	8.7	2.6	0.0	0.9
0.0	0.0	0.6	9.1	0.6	1.6	0.0	5.8	0.0	1.3	0.0	73.1	1.3	0.6	0.0	0.6	3.9	0.0	1.0	0.3
0.0	0.0	0.0	13.1	0.5	1.6	0.0	0.0	0.0	0.0	0.0	3.8	76.0	1.6	0.0	0.0	1.6	0.5	0.5	0.5
0.2	0.1	0.3	0.3	0.2	0.6	0.1	0.3	0.1	1.8	0.3	0.1	0.0	92.1	0.1	0.1	1.8	1.0	0.2	0.1
0.0	0.0	0.5	1.1	0.0	1.1	3.2	5.3	0.0	6.3	0.5	0.0	0.0	1.6	75.7	0.5	2.1	1.6	0.0	0.5
0.0	1.0	0.0	0.0	3.1	13.3	4.1	2.0	3.1	10.2	0.0	1.0	0.0	5.1	0.0	44.9	12.2	0.0	0.0	0.0
0.2	0.5	3.3	0.5	0.3	4.1	0.0	0.4	0.5	3.2	0.3	0.2	0.1	3.2	0.1	0.7	81.6	0.3	0.2	0.0
0.9	0.0	0.5	0.5	0.0	12.4	1.4	0.5	0.5	20.6	0.0	0.0	0.0	10.1	1.4	0.0	2.8	47.4	0.5	0.5
0.8	0.0	0.0	5.4	1.4	0.0	0.0	2.8	0.0	0.8	0.0	0.3	0.3	1.1	0.3	0.0	1.4	0.0	75.5	9.9
0.6	0.6	0.0	11.0	1.7	1.1	0.0	0.6	0.0	1.1	0.0	4.4	2.2	2.2	0.0	0.0	0.6	0.6	20.4	53.0

VI. Conclusion and Future work

Throughout the project, the team made five different feature sets including the original feature set, and put them into tests with OVA Naive Bayes, SVM, Logistic regression, SVM by Crammer and Singer and K nearest neighbor.

The result shows that logistic regression yields the best performance.The team believed that the reason logistic regression has the highest accuracy is related with the fact that logistic regression performs well with large training set with non-Gaussian distribution.

Even though it has the highest accuracy, the performance of logistic regression is closed to the performance of the two SVM classifiers compared with the performance of Naive Bayes and K nearest Neighbor. It is make sense that SVM tends to yield better performance since it is a robust algorithm, the fact the different feature sets doesn't change the performance too much is probably because changing the features doesn't effect the forming of support vectors a lot.

The confusion table shows that some classes are highly correlated. This correlation reduces the accuracy of the algorithm. Given more time, the team would investigate deeper into the correlation and came up with way to better distinguish these specific classes.

The table also indicates that the classes error rate is also influenced by the number of instances of these classes in the training set. Since the training set is not uniformly distributed with each class, the team would find a better way to balance them, since simply duplicating the classes with small instance size didn't yield better result.

References

[1] J. Rennie, R. Rifkin, "Improving Multiclass Text Classification with the Support Vector Machine, " Revised, April 2002.

[2] K. Crammer, Y. Singer, "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines, " *Journal of Machine Learning Research*, 2001.

[3] D. Trudgian,Z. Yang, "Spam Classification Using Nearest Neighbour Techniques, " *Proceedings of the 5th international conference on intelligent data engineering and automated learning*, pp.578–585.Revised, 2004.

[4] Yang, Yiming, Pedersen, Jan O, "A comparative study on feature selection in text categorization, " *ICML*, vol. 97, pp. 412-420, 1997.

[5] Ko, Youngjoong, "A study of term weighting schemes using class information for text classification, " *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, vol. 97, pp. 1029-1030, 2012.

[6] Yang, Yiming, and Xin Liu, "A re-examination of text categorization methods, " *Proceedings of the 22nd international ACM SIGIR conference on Research and development in information retrieval*, ACM, 1999.