Leaflets Three, Let it be?
—Mushroom Edibility Classification

ST 599 Big Data
Project 3

Edited by

Shangjia Dong
Martin Guyer
Wanli Zhang

1 Introduction

Mushroom edibility is determined by many different attributes. Conducting a poison test every time before eating is not realistic. Therefore, finding a method that enables us to judge the edibility by looking at its physical properties like color, shape, habitat etc., is essential. In this study, we use classification methods to develop a rule for differentiating edible mushrooms from poisonous ones. The data set includes hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family with 8124 instances and 22 attributes. We noticed 2480 missing values in the "Stalk Root" attribute, and after further investigation, we found out that they were not missing completely at random. Therefore, we decided to treat the missing values as another category of this attribute.

2 Machine Learning Method

Due to the categorical nature of data, the number of classification algorithms available is rather limited. Eventually, we chose the **naive Bayes classifier** as the one to work with. In general, Bayesian classifiers assign the most likely class to a given instance described by its feature vector, and naive Bayesian classifiers impose a strong (naive) assumption that features are independent given class. Given feature variables $F_1, ..., F_n$, a Bayesian classifier aims at computing the probability

$$p(C|F_1, ..., F_n)$$

where C denotes a class variable with some number of classes.

Using Bayes' theorem, we have:

$$\begin{aligned} p(C|F_1, ..., F_n) &= \frac{p(C)p(F_1, ..., F_n|C)}{p(F_1, ..., F_n)} \\ &\propto p(C, F_1, ..., F_n) \end{aligned}$$

Using the chain rule on the right-hand side, it follows that:

$$\begin{aligned} p(C|F_1, ..., F_n) \propto\ & p(C)p(F_1|C)p(F_2|C, F_1) \\ & \cdots p(F_n|C, F_1, ..., F_{n-1}) \end{aligned}$$

Since F_k's are mutually independent conditioning on class C, the joint model above becomes:

$$p(C|F_1, ..., F_n) \propto p(C) \prod_{i=1}^{n} p(F_i|C)$$

and the exact value of the prediction probability is then:

$$p(C|F_1, ..., F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

where $Z = p(F_1, ..., F_n)$ is free from class C and only depends on values of the feature vector.

The class prior $p(C)$ and feature probability distribution $p(F_i|C)$ can be approximated from the training set. A class prior can be calculated by assuming equiprobable classes, or directly estimated by relative frequencies in the sample. The approximation of the feature distribution, however, requires us to impose a distribution on the features in the training set. In this study, each feature is either a binary or multi-class categorical variable. Therefore, the Bernoulli (for binary) and multinomial (for multi-class) distributions are natural choices for feature distribution.

After prediction probabilities are acquired, a classification rule is needed to assign class labels to new instances. Sometimes, the *maximum a posteriori* (MAP) decision rule is used, in which an instance is placed in the most probable group. However, when the problem is a binary classification, a threshold is often used to determine the class label: if the prediction probability falls above the threshold, the instance is labeled positive, and if not, negative.

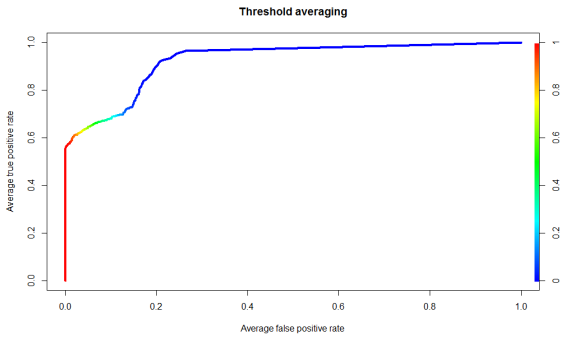
To assess the performance of this classifier in practice, we applied a K-fold cross validation: the data set is split into six equally-sized subsets and we applied naive Bayes six times, each time leaving out a different subset to be tested while using the remaining five as the training set. At first, we used the *Receiver Operating Characteristics* (ROC) curve to select an optimal threshold to map instances to predicted classes. Then we constructed confusion matrices and computed the *Apparent Error Rate* (APER), defined as the fraction of misclassified sample observations, for each run of cross validation as well as their average. Finally, we calculated the *False Negative* (FN) rate for each run, which, in this study, is defined as the probability of classifying a poisonous (positive) mushroom as being edible (negative), because this situation leads to more deadly consequences than getting a false positive and hence is our major concern.

3 Findings

3.1 ROC Curve

A ROC graph is a technique for visualizing and selecting classifiers based on their performance (Fawcett, 2005). In this study, we used it to select the optimal threshold for prediction probabilities. ROC curves are two-dimensional graphs in which true positive (TP) rate is plotted against false positive (FP) rate. An ROC graph depicts relative tradeoffs between benefits (TP) and costs (FP). Ideally, an optimal classifier is one that identifies all true positives and no false positives. Therefore, roughly speaking, the more "northwest" a point lies in ROC space, the better the corresponding classifier performs.

For a probabilistic classifier such as naive Bayes, each point in ROC space corresponds to a different choice of threshold. We constructed a ROC curve for each fold of cross validation and averaged the curves at each threshold. The resulting graph is as follows:



Notice that according to the color palette, the upper left corner (marked by a cross) corresponds to threshold values close to zero.

3.2 APER and False Negative Rate

Based on the ROC plot above, we set the threshold to 0.005, which is close to zero. Using this classification rule, the APER and FN rate of each run are as follows:

Run	APER	FNR
1	0.0798	0.0392
2	0.0214	0.2302
3	0.0805	0.2013
4	0.0517	0.0234
5	0.0805	0.0428
6	0.2349	0.0385
avg	0.0798	0.0392

Therefore, on average, with our choice of threshold, the naive Bayes classifier mistakenly classifies 7.98% of mushrooms, and 3.92% of poisonous mushrooms are wrongfully identified as being edible.

4 Discussion

4.1 Assumptions

As we mentioned in Section 2, the main assumption of naive Bayes is independence between features given class labels. Although this assumption is unrealistic in practice, it has been shown that naive Bayes may still have high rates of accuracy for data sets in which strong dependencies exist among attributes (Zhang, 2004). The reason behind the classifier's robustness still remains an open question.

4.2 Scaling to Big Data

The data set we worked with contains 8124 observations and 22 independent variables. The actual running time of the six-fold cross validation is 6.13 seconds, and we expect this time to increase at least linearly with sample size of the validation set, as well as the number of features: Suppose a new categorical variable with k levels is added, then if we assume a multinomial distribution, an extra $k-1$ parameters need to be estimated.

References

[1] Fawcett, T. (2005): "An introduction to ROC analysis", *Pattern Recognition Letters*, 27, 861-874

[2] Zhang, H. (2004): "The optimality of naive Bayes", FLAIRS2004 conference