UASimpleClassifier: An implementation of Naive Bayes Classifier

Noah Buchanan
Computer Science Department
University of Arkansas - Fort Smith

February 21, 2021

Abstract

Naive Bayes classifier is a powerful classification tool that simplifies the classification process through ignoring feature dependencies and assuming all features are independent. My goal was to create a algorithm that utilizes the power of classification through naive Bayes and and to do research of how others have gone about the process of naive Bayes and highlight the big points.

Introduction/Background

This is my implementation of a naive bayes classifer, UASimpleClassifer. Using probabilities of features, classes, and features given classes as specified in the bayes formula to classify categorical and also continuous data alike.

Implementation/Technique

The core technique for this implementation is using naive Bayes to train the model, calculate probabilities based on the features for the classes provided and classify new data under this model. For this to work we need to store the data in a way that we can access the probability of the individual classes, the probability for each individual discrete feature given a class and account for continuous features as well given a class, in this case feature 4 and feature 5. My implementation uses data structures of type:

$$ArrayList < String >$$

This data structure simply stores the class ”keys” needed to access the data related to a specific class

$$HashMap < String, HashMap < Integer, ArrayList < String >>>$$

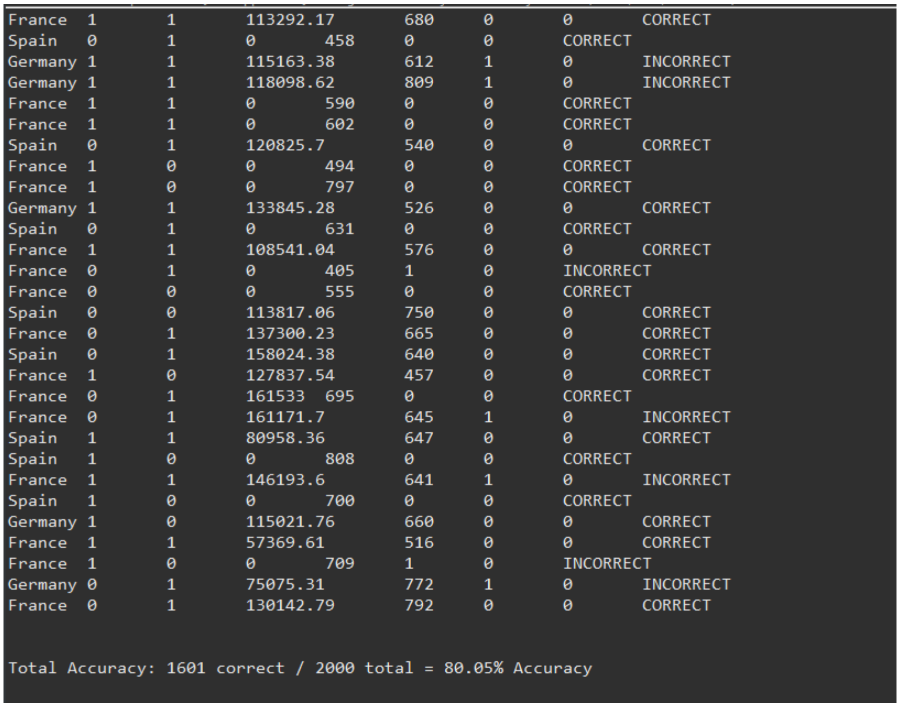
The purpose of this one is storing the individual values of the features based upon the feature they come from and the features they come from are filtered once again by the class they belonged to when training the model. For instance lets call the data structure ”data” for this example: data.get(”class1”).get(1).get(0) will retrieve the first value for feature 1 that we found that belonged to ”class1” when training the model.

$$HashMap < String, ArrayList < Integer >>$$

This one is simply for the sole purpose of keeping track of the amount of values found for the specific classes for calculating the probabilities of the specific classes.

The train method simply stores all the data sorted by class and feature for the training data that we will be using. The test method uses the stored data and attempts to classify the test data given based on the probabilities of the training data using Naive Bayes Classifier. The classify method is what does the actual probability calculations and final classification it calls a various amount of other methods to achieve this which have been included in the source code.

Evaluation



```
France  1       1       113292.17       680     0       0       CORRECT
Spain   0       1       0       458     0       0       CORRECT
Germany 1       1       115163.38       612     1       0       INCORRECT
Germany 1       1       118098.62       809     1       0       INCORRECT
France  1       1       0       590     0       0       CORRECT
France  1       1       0       602     0       0       CORRECT
Spain   0       1       120825.7        540     0       0       CORRECT
France  1       0       0       494     0       0       CORRECT
France  1       0       0       797     0       0       CORRECT
Germany 1       1       133845.28       526     0       0       CORRECT
Spain   0       1       0       631     0       0       CORRECT
France  1       1       108541.04       576     0       0       CORRECT
France  0       1       0       405     1       0       INCORRECT
France  0       0       0       555     0       0       CORRECT
Spain   0       0       113817.06       750     0       0       CORRECT
France  0       1       137300.23       665     0       0       CORRECT
Spain   0       1       158024.38       640     0       0       CORRECT
France  1       0       127837.54       457     0       0       CORRECT
France  0       1       161533  695     0       0       CORRECT
France  0       1       161171.7        645     1       0       INCORRECT
Spain   1       1       80958.36        647     0       0       CORRECT
Spain   1       0       0       808     0       0       CORRECT
France  1       1       146193.6        641     1       0       INCORRECT
Spain   1       0       0       700     0       0       CORRECT
Germany 1       0       115021.76       660     0       0       CORRECT
France  1       1       57369.61        516     0       0       CORRECT
France  1       0       0       709     1       0       INCORRECT
Germany 0       1       75075.31        772     1       0       INCORRECT
France  0       1       130142.79       792     0       0       CORRECT


Total Accuracy: 1601 correct / 2000 total = 80.05% Accuracy
```

The accuracy of my implementation given 3 categorical features and 2 non-categorical was 80% on average given different splits of data but all followed the trend of 80/20, 80% training 20% testing. There were 10,000 records that were being split 80/20, and with 2 classes to classify from 80% accuracy is far higher than the flipping a coin odds we would have without the classification algorithm.

Conclusion

In summary my implementation predicts the classification of data with 80% accuracy, keep in mind this was ignoring feature dependency as naive Bayes does however the results were still quite successful. The uses for predicting classes are numerous, determining spam emails with high accuracy, predicting market trends based on training data from different types of market trends in the past. You could even predict stock changes if you factored in enough data into your training and testing.

Relevant research

There's lots of research already that parallels what I have done here in this implementation and I would like to make note of some similarities and note-worthy points. Something that I thought quite interesting regarding the naivety of the the implementation that parallels mine own algorithm, despite its unrealistic independence assumption, the naive Bayes classifier is surprisingly effective in practice because it is often correct even when the probability estimates are inaccurate [1]. Naive Bayes is typically ineffective in the case of binary features however despite its limitation naive Bayes has show to be optimal for some important classes of concepts that have a high degree of feature dependencies such as disjunctive and conjunctive concepts [1] which would seem to not make any sense at first glance yet is a result of naive Bayes nonetheless. Another paper of similar concept as we just learned naive Bayes assumes feature independence and has a high accuracy classification rate even for binary features, however in the case of non-binary but also categorical features it operates are particularly high rate of accuracy [2]. They showed a multi-nomial model rather than a bi-nomial model(binary features) operated by an average of 27% lower and sometimes by more that 50% lower error [2]. Those are extremely impressive results in comparison and I think extremely note-worthy for determining the right situations to use naive Bayes as a classifier.

References

[1] I. Rish *et al.*, “An empirical study of the naive bayes classifier,” in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, 2001, pp. 41–46.

[2] A. McCallum, K. Nigam *et al.*, “A comparison of event models for naive bayes text classification,” in *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1. Citeseer, 1998, pp. 41–48.