Implementing and Testing Winnow-2 and Naive Bayes From Scratch
By: Jonathan Klinger

Abstract:
The purpose of this paper is to show my thought process in both coding the Winnow-2 and Naive Bayes algorithms from scratch and using 3 datasets to test the performance of these models. I chose to use the breast cancer dataset, iris dataset, and the house votes dataset. The results were as follows: 1) for the breast cancer dataset, naive bayes performed slightly better with an accuracy of 97% vs. Winnow's 96%, 2) for the iris dataset, naive bayes performed better with an accuracy of 74% vs. Winnow's 70%, but both results were low because of the approach I took to binary multi-class classification, 3) for the house votes dataset Winnow-2 performed slightly better with an accuracy score of 94% vs. Naive Bayes' score of 93%. It is worth noting that when using a multi-class approach on the iris dataset, I received an accruacy of 98%.

The task at hand was to show how my model's performed on three different datasets. My hypothesis was that Naive Bayes would perform much better than Winnow-2 on all datasets. This is because the Winnow-2 more simplistic, using weight updates to train the classifier. Naive Bayes is much more complex using frequencies and maximum likelihood estimation for classification. Additionally, Naive Bayes is used much more in the real world, whereas Winnow-2 is more of an academic model. I also suspected that in cases where a dataset had real-valued data, Naive Bayes would be more accurate because it can handle the data using the Gaussian approach. The Winnow-2 algorithm can only handle 1s and 0s for each feature, so we are forced to one-hot encode them and lose the relationships between the numbers.

Implementation of Algorithms:
The Gaussian Naive Bayes algorithm is implemented as follows: a fit( ) function that takes in the training data and calculates the frequency for each class (the priors), the mean for each feature given the class, and the variance for each feature given the class. We also have a predict( ) function that for each class: 1) takes the log likelihood (assuming the distribtion is Gaussian) of each feature given the class using the pdf function, 2) sums up each of those conditional probabilities (because we have taken the log), 3) adds the prior and the sumed conditional probabilities). After that the model takes the argmax of the posteriors for each sample, and classifies the data with the corresponding label.

The Bernoulli Naive Bayes algorithm is implemented as follows: a fit( ) function that calculates the frequency of each class and the conditional probability of each feature given the class. A predict class that takes in the testing data and for each class does the following: 1) given the class, calculates the log likelihood of each feature value given the class, using the Bernoulli distribution, 2) sums the conditional probabilities for each feature, 3) adds the prior and the conditional probabilities for each class to get the posterior. Once we have the posterior for each class, we take the argmax for each sample, and classify the data with the corresponding label.

The Winnow-2 algorithm is implemented as follows: a fit( ) function (with a default threshold of 0.5 and a learning rate of 2) that takes in the training data and initializes each features weight with 1. For each sample, the function multiplies each feature with the corresponding weight and sums each of the resulting values. If the result is higher than the threshold then it categorizes it as a 1, if not 0. If the prediction is correct, the function stops there. If it is incorrect, it checks to see if it should promote or demote the weights. The promote( ) function multiplies the weight by the learing rate if the value of that feature was 1 and does nothing if it is 0. The demote function divides the weight by the learning rate if the value of that feature was 1 and does nothing if it is 0. The predict( ) function iterates through the test data using the weights we have trained on, performing the same process of multiplying each feature value by the weights and summing them, it checks to see if the result is higher than the threshold. If it is, the function classifies the sample as 1. If it is lower than the threshold, the function classifies the sample as 0.

Approach:
The breast cancer and house votes dataset were both missing data. The missing features from the breast cancer data were real-values and the missing data from the house votes dataset were categorical. To impute these, I created an impute function that takes the mean of the missing feature given the class for real-valued data and rounded the mean to 0 or 1 for the categorical data. To do this for the house votes dataset, I needed to change the feature values to 1 and 0 for "y" and "n" respectively. I also changed the labels for each dataset to 0 and 1 for computational purposes.

For the breast cancer and iris dataset, I needed to one-hot encode each feature so that they can be used in the Winnow-2 algorithm. After that, the preprocessing for the breast cancer dataset was complete and I ran the Gaussian Naive Bayes and Winnow-2 algorithm on the data.

The iris dataset had three classes, which was a problem for the Winnow-2 algorithm. To combat this, I created three seperate versions of the data. Each version augmented the data to facilitate a binary classification for each of the three classes. I augmented the Gaussian Naive Bayes model to output the posteriors and the Winnow-2 model to output the distance from the threshold for each sample. After that, I took the class with the max posterior and the max distance from the threshold to be the true class of the sample.

The house votes dataset is when I realized why it was recommended to not use the Gaussian Naive Bayes model. I ran the model on the data and received extremely low performance. I quickly realized that I needed to use Bernoulli Naive Bayes because the data is categorical and only takes on 2 values (0 and 1). I created the Bernoulli Naive Bayes model and got the results. The winnow-2 algorithm handled this data very easily because the features are already 0 or 1.

Results (results are different than in the video… these were computed on multiple runs):

	Naive Bayes	Winnow-2
Breast Cancer	97%	96%
Iris	88%	82%
House Votes	94%	93%

Behavior of algorithms:
The results were interesting. As I hypothesized, Naive Bayes consistently outperformed for each dataset. I was interested in seeing why the performance was so low for the Iris dataset. I suspected that the reason for this has something to do with handling the three classes by making the problem a multi-class binary classification. I did some digging and found an academic paper that concluded: "Naive Bayes with one-against-all binarization is not equivalent to a regular Naive Bayes"[1] The paper alludes to the added class imbalance as the source for this non-equivalence. To combat this, we would have to use a pairwise Naive Bayes classifier. To confirm my suspicions, I tested the iris data using multi-class classification and found received an accuracy of 98%.

I also found it interesting that Winnow-2 performed so well on the House Votes dataset. My hypothesis is that the performance is due to the feature types. The 0/1 features are exactly what Winnow-2 requires to perform well. We didn't have to manipulate the features at all, so there is no loss in accuracy due to the manipulation in the data. Winnow-2 was made to handle a problem like this.

Given the above, I was surprised to see that Winnow-2 performed as well as it did on the breast cancer dataset. The features were numerical and continuous. We had to round the data and one-hot encode it to run it through Winnow-2, so there is a loss in the ordinal qualities of this numerical data. To look into this suspect performance, I calculated the means for each featue given the label. The means for each feature were very far from one another, so one-hot encoding won't lose too much accuracy on average. This is because the 1s for cancer in the one-hot encoded feature are mostly on one side of the vector, while the 1s for no-cancer are on the other side.

Summary:
The performance for the breast cancer and house votes dataset were very strong. Although Winnow-2 isn't used in practice very much it performed almost as well as Naive Bayes. For Winnow-2, the results were likely due to the highly differentiated features for the breast cancer dataset and the type of features fitting well with the nature of Winnow-2 for the Winnow-2 algorithm. The lower results for the Iris dataset were due to my using a one-against-all approach to the binary multi-class classification. This created a class imbalance that hindered the performance. I conclude this by testing with a normal Naive Bayes model (mult-class) and receiving 98% accuracy.

Sources:

[1] On Pairwise Naive Bayes Classifiers by Jan-Nikolas Sulzmann Johannes Fürnkranz
Eyke Hüllermeier