PROJECT 1 - WINNOW2 AND NAIVE BAYES

DAVID ATLAS

Abstract. This paper will introduce the Winnow2 algorithm and the Naive Bayes algorithm. Both will be treated in context of classification problems, with strategies for categorical and continuous valued inputs. In several experiments on real world datasets, both algorithms perform comparably, with Naive Bayes at a slight advantage on high dimensional continuous feature spaces, and Winnow2 at a slight advantage with on high dimensional boolean feature spaces.

1. Problem Statement & Hypothesis

Two basic supervised learning methods are introduced, and applied to real world classification problems. The first, Winnow2, is a simple learning model for boolean inputs and outputs. The second, Naive Bayes, is a simple graphical model that can handle multivalued discrete and continuous inputs.

Both models will be compared using boolean valued inputs for comparability, while Naive Bayes will also be applied to multivalued discrete and continuous inputs, testing for general superiority in those spaces.

Both algorithms do not explicitly treat interactions between variables (Winnow2 will adjust the weights for all the active weights, and Naive Bayes assumes the feature values are conditionally independent on the class), so large differences in performance between the two are not expected. However, when Naive Bayes is given the capacity to treat continuous inputs via a Gaussian conditional distribution, and the inputs do not easily lend to manual segmentation of the classes, it is expected that Naive Bayes may learn a better representation of the function.

Additionally, given the simplicity of the algorithms, it might be reasonable to expect Winnow2 to perform well on large feature spaces, as it has an implicit regularization mechanism (weights are only updated if boolean features are active). Naive Bayes may struggle in those large spaces with overfitting. This issue might also extend to class imbalances, as there won't be many training instances over which to estimate the conditional distributions.

In summary, the hypothesis is that of rough performance parity, with Winnow2 having an advantage in large feature spaces, and Naive Bayes having an advantage with continuous features. In accordance with the No Free Lunch theorem, it would be unrealisitc to expect one algorithm to perform better on all problems.

2. Description of Algorithms

Winnow2. The Winnow2 Algorithm[4] is a linear-threshold boolean classifier, in which a set of boolean inputs is mapped to a single boolian output. The algorithm works as follows:

Given a boolean input vector, $X = (x_1, \cdots, x_k)$, initialize a vector of weights $W = (w_1, \cdots, w_k)$. We also initialize two parameters, $\theta > 0$ and $\alpha > 1$. For each training instance, we calculate $Z(X, W) = \sum_{i=1}^{k} x_i w_i$. If $Z \geq \theta$, we predict a 1, and if $Z < \theta$, we predict a 0.

The learning mechanism is composed of two operations - promotion and demotion. Under promotion, w_i is updated such that $w_i = \alpha w_i \ \forall i \mid x_i = 1$. Under demotion, w_i is updated such that $w_i = \frac{w_i}{\alpha} \ \forall i \mid x_i = 1$. If the prediction is a 1, while the correct response is a 0, demotion is applied. If the prediction is a 0 while the correct response is a 1, promotion is applied. This has the effect of raising the value of Z when the function predicts a value that is too small, and lowers the value of Z when the function predicts a value that is too large.

If a given training example yields a correct prediction, no updates to the weight vector are made. This is the regularizing effect mentioned above.

As mentioned in [4], this algorithm is an online learner, insomuch as given a set of weights, and a training example, the algorithm can make the update without awareness of the other training examples, allowing it to train continuously on new examples coming in without any adjustment.

Naive Bayes. The Naive Bayes algorithm[1] is a probablistic graphical model that leverages Bayes' Rule and some simplifying assumptions. To recap, under Bayes' Rule:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}. \tag{1}$$

Suppose C is the random variable representing the class of an instance, while X is a set of random variables representing feature values of a given instance. In creating a classifer, a reasonable goal can be calculating the likelihood that a given instance belongs to class C, given its feature values X. As such, given the right hand of Equation 1, the goal would be achieved.

In the case of a high dimensional feature set $X = (x_1, \cdots, x_k)$, Equation 1 is

$$P(C \mid X_i, \cdots, X_k) = \frac{P(X_i, \cdots, X_k \mid C)P(C)}{P(X_i, \cdots, X_k)}$$

To make the equation above more managable, a simplifying assumption can be made - suppose all of the features are independent conditional on the class. Then for a set of k feature values,

$$P(C \mid X_i, \cdots, X_k) = \frac{P(X_i, \cdots, X_k \mid C)P(C)}{P(X_i, \cdots, X_k)} = \frac{\prod_{i=1}^{k} P(X_i \mid C)P(C)}{P(X_i, \cdots, X_k)} \tag{2}$$

This simplifying assumption is often unrealistic, as feature values are likely not independent conditional on the class. However, in [1], empirical evidence is cited that Naive Bayes performs well relative to more complex learning algorithms that account for potential dependence relationships between features, given the class.

Returning to the algorithm, the classification function can be defined as choosing the value of C that is most likely, given the feature values, and by extension:

$$\arg\max_C P(C \mid X_i, \cdots, X_k) = \arg\max_C \prod_{i=1}^{k} P(X_i \mid C)P(C).$$

The denominator in Equation 2 was dropped, as its value is not impacted by the class of a given instance, and so it won't affect the $\arg\max_c$ operator.

With that background, $P(X_i \mid C)$ and $P(C)$ must be estimated. If X_i is a discrete random variable, $P(X_i \mid C)$ can be described as multinomial, where p is the proportion of each value of X_i for training instances of class C. If X_i is continuous, $P(X_i \mid C)$ can be described as Gaussian, with a mean of the sample mean of X_i for training instances of class C and a standard deviation of the sample standard deviation of X_i for training instances of class C.

$P(C)$ is multinomial, with p equal to the proportion of total training examples of class C. $P(C)$ would be binomial if it is boolean under this model.

Note that all of these distributional family assumptions can be changed if appropriate, but are generally sensible defaults where no other distributional information is known.

Smoothing can be applied in situations where there are no training examples of class C that take on a value $X_i = Q$ (where Q is a constant), to impose a non-zero likelihood for $P(X_i \mid C)$, which can lead to better generalization. For a multinomial distribution, $p_i = \frac{\sum_{i \in C} x_i + \alpha}{\sum_{i=1}^{n} x_i + \alpha m}$, where $\alpha = 1$ and m is the number of values in X_i. Obviously, this can be done in many other ways, but this is a sensible default. For a continuous distribution, the distribution over all classes can be used as a prior.

After calculating the conditional distributions of the training examples, the prediction is simply

$$\arg\max_C \hat{P}(X_i \mid C)\hat{P}(C),$$

where $\hat{P}$ indicates the estimated distribution from the training process.

3. Experimental Approach

The experiments conducted included 5 datasets:

(1) The iris dataset classifying plant species from leaf measurements.
(2) The breast cancer dataset classifying tumors based on breast measurements.
(3) The glass dataset classifying the origin of broken glass based on measurements of the shards.
(4) The soybean dataset classifying rot based on crop information.

	Range 1	Range 2	Range 3
Sepal Length	[0, 5.5]	(5.5, 10]	
Sepal Width	[0, 3]	(3, 10]	
Petal Length	[0, 2]	(2, 5]	(5.5, 10]
Petal Width	[0, 1]	(1, 1.6]	(1.6, 10]

TABLE 1. Discretization bins for the Iris Dataset, based on Figure 1

	Setosa	Versicolor	Virginica
Setosa	9	0	0
Versicolor	0	8	0
Virginica	0	0	13

TABLE 2. Iris - Winnow2 with boolean inputs confusion matrix

	Setosa	Versicolor	Virginica
Setosa	9	0	0
Versicolor	0	8	1
Virginica	0	0	12

TABLE 3. Iris - Naive Bayes with boolean inputs confusion matrix

	Setosa	Versicolor	Virginica
Setosa	9	0	0
Versicolor	0	8	1
Virginica	0	0	12

TABLE 4. Iris - Naive Bayes with Continuous Inputs Confusion Matrix

(5) The congressional voting dataset classifying party based on legislation votes.

For each dataset, discrete inputs were one-hot encoded as booleans, and continuous values were discretized based on boxplots of their distributions, segmented by class, to attempt to provide one-hot encodings that best seperated the classes. For multivalued outputs, one vs. rest classifiers were fit. For Winnow2, the same threshold θ was used for all the classes, and the highest value of Z was selected across each classifier. For Naive Bayes, the highest likelihood class was selected.

For each dataset, Winnow2 and Naive Bayes algorithms were used on boolean inputs and outputs for comparability. Additionally, Naive Bayes was fit using multivalued discrete and continuous attributes. A 5 fold randomly-shuffled cross-validation approach was used to evaluate the models.

4. Experimental Results

4.1. **Iris Dataset.**

Data Cleaning & Transformation. The first dataset presented is the Iris dataset[2]. First, the continuous inputs are discretized. Figure 1 was used to pick the cutoff points by finding values that best seperate the classes in a univariate sense. The following cutoff points were used: One-hot encodings were used on the discretized features.

Model Fitting Results. The model results for one of five random folds are shown in Tables 2, 3 and 4. Note that for the confusion matrices here and below, the column headers are the actual labels, while the row headers are the predicted values. Values along the diagonal indicate correct predictions. Winnow2 scores a 100% accuracy rate, which both forms of Naive Bayes get 96.6% accuracy.

The model results for Naive Bayes with boolean inputs are shown in Table 4. These results indicate that the classes are easily seperably in a 4D space of all the features. Both algorithms perform quite well, with Winnow2 classifying everything correctly, and Naive Bayes missing just one test example. However, note that Winnow2 is aided here by the manual discretization of the boundaries between classes.

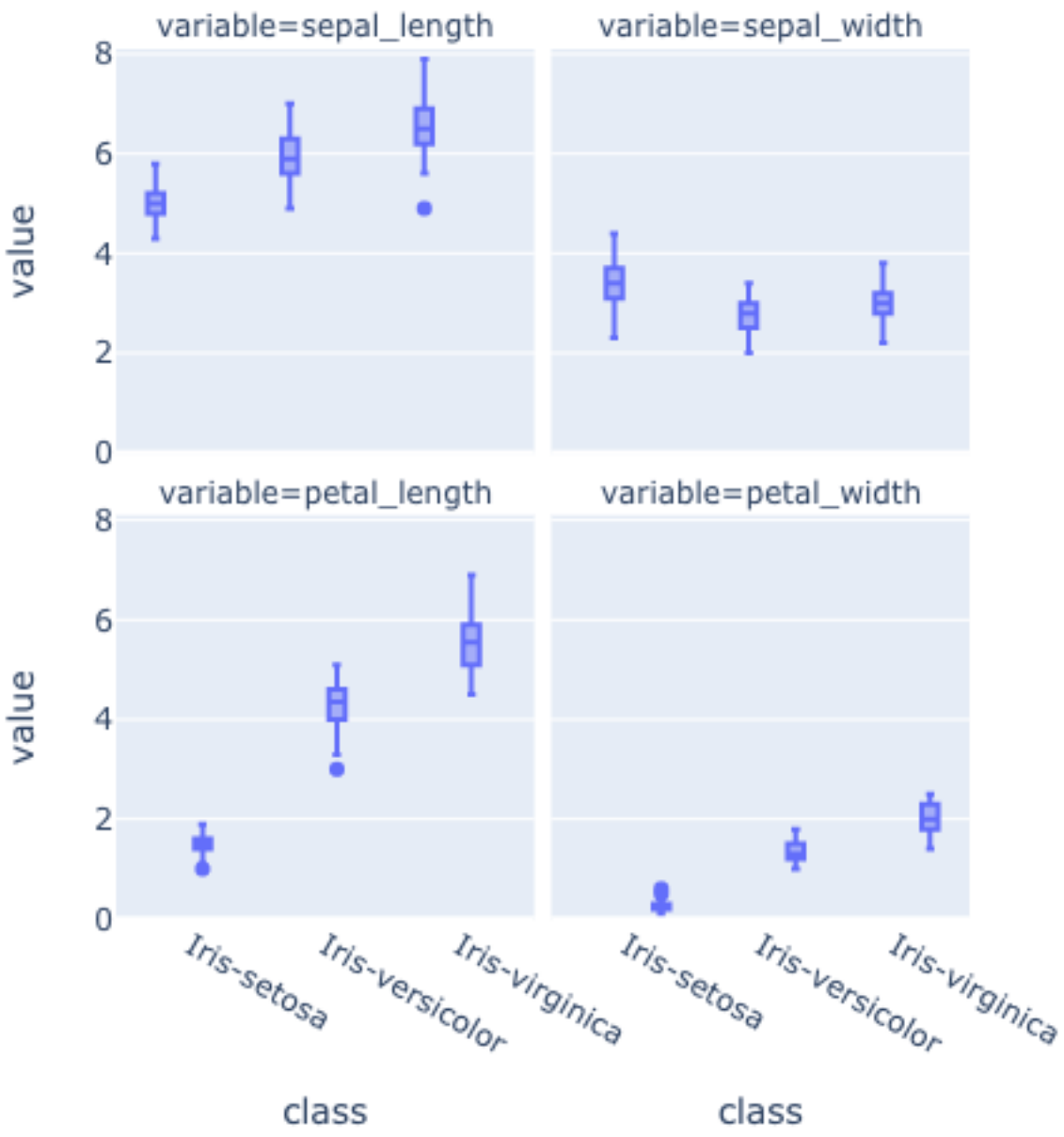

FIGURE 1. Iris dataset boxplot of feature values by class.

4.2. **Cancer Data.**

Data Cleaning & Transformation. For the cancer dataset[7], all null values are dropped, as this is only a small proportion of total rows (16/699). The inputs are multivalued discrete, so one-hot encoding representations are made of each column.

Model Fitting Results. The model results for Winnow2 with one-hot encoding variables are shown in Table 5. The model results for Naive Bayes with one-hot encoding variables are shown in Table 6. The model results for Naive Bayes with multinomial input variables are shown in Table 7. The results here indicate near performance parity. Winnow2 does quite well, as this problem has many attributes, only some of which are

	Benign	Malignant
Benign	86	0
Malignant	0	50

TABLE 5. Cancer - Winnow2 results.

	Benign	Malignant
Benign	84	1
Malignant	2	49

TABLE 6. Cancer - Naive Bayes with boolean inputs results.

	Benign	Malignant
Benign	83	0
Malignant	3	50

TABLE 7. Cancer - Naive Bayes with multivalues discrete inputs results.

Class	refractive index	sodium	magnesium	aluminum	silicon	potassium	calcium	barium	iron
0	1.518	13.242	3.552	1.163	72.619	0.447	8.797	0.012	0.057
1	1.518	13.111	3.002	1.408	72.598	0.521	9.073	0.050	0.079
2	1.517	13.437	3.543	1.201	72.404	0.406	8.782	0.008	0.057
3	1.518	12.827	0.773	2.033	72.366	1.470	10.123	0.187	0.060
4	1.517	14.646	1.305	1.366	73.206	0.000	9.356	0.000	0.000
5	1.517	14.442	0.538	2.122	72.965	0.325	8.491	1.040	0.013

TABLE 8. Glass Data - Feature means by class. This is used to inform the discretization of the features.

likely relevant. The inputs are not continuous, and so Naive Bayes does not have a significant advantage. The exact accuracy numbers are Winnow2: 100%, Naive Bayes (Boolean Inputs): 97.8%, Naive Bayes (Multinomial): 97.8%.

4.3. **Glass Data.**

Data Cleaning & Transformation. The glass dataset[3] features are continuous, and so the same procedure as above is followed to discretize them. Table 8 shows the mean by class for each of the features. The values chosen reflect points that seperate the means of the classes reasonably. The values chosen are shown in Table 4.3. One-hot encodings are made for each of the discrete cutoff points.

Model Fitting Results. The confusion matrix is shown in Table 4.3 for the Winnow2 algorithm, using the discretization points above. Naive Bayes was also run with the boolean inputs, and the results are shown in Table 4.3. Naive Bayes was then run with the continuous inputs, and the results are shown in Table 4.3 This experiment results in accuracy of 40% for Winnow2, 43% for Naive Bayes with boolean inputs, and 45% for Naive Bayes with continuous inputs. This fits with the hypothesis described above - performance is roughly similar with Naive Bayes having an advantage in continuous spaces. Note that the accuracy numbers here are much lower than in the other problems, some of which can potentiall y be attributed to that higher number of classes.

4.4. **Soybean Data.**

Data Cleaning & Transformation. The next dataset is the Soybean dataset[5] in which the rot associated with a soybean is predicted based on various attributes of the soybean. The features are all multivalued discrete, but some only have a single value. Columns with a single value are dropped, as they cannot help learn classes. The rest of the classes are then mapped into one-hot encodings.

	Range 1	Range 2	Range 3	Range 4
refractive index	1.518			
sodium	12.500	14.0		
magnesium	1.000	2.0	3.25	
aluminum	1.200	1.5	2.00	
silicon	72.600	72.8	73.20	73.4
potassium	0.200	0.4	0.50	0.6
calcium	8.600	9.0	10.00	
barium	0.350			
iron	0.200	0.6		

	0	1	2	3	4	5
0	13	7	4	1	0	3
1	0	3	0	2	0	0
2	0	5	1	0	0	3
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

TABLE 9. Glass dataset Winnow2 results.

	0	1	2	3	4	5
0	8	2	4	1	0	1
1	5	10	1	2	0	5
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	2	0	0	0	0

TABLE 10. Glass dataset Naive Bayes boolean input results.

	0	1	2	3	4	5
0	7	11	1	0	0	0
1	0	1	0	0	0	0
2	6	2	4	0	0	0
3	0	1	0	2	0	1
4	0	0	0	0	0	0
5	0	0	0	1	0	5

TABLE 11. Glass dataset Naive Bayes continuous input results.

Model Fitting Results. All of the algorithms perform extremely well on the soybean dataset, with all 3 variants classifying all test set examples correctly. The confusion matrix for all three is shown in Figure 4.4.

4.5. **House Votes.**

Data Cleaning & Transformation. The House Votes dataset[6] has the boolean values for whether a congressperson voted for or against a measure, and their party affiliation. For the House Votes dataset, all inputs are boolean. There are missing values scattered throughout the dataset, and so when the data is encoded as a boolean value, null values will be considered their own value, and get their own one-hot encoding.

Model Fitting Results. The Winnow2 results are shown in Table 4.5, with 96.5% accuracy. The Naive Bayes results are shown in Table 4.5 with 91.9% accuracy. A second iteration of the Naive Bayes model was not fit, as the dataset was simply a set of booleans, and so fitting another Naive Bayes would simply be equivalent.

	D1	D2	D3	D4
0	2	0	0	0
1	0	1	0	0
2	0	0	2	0
3	0	0	0	4

TABLE 12. Soybean data models all correctly classify all examples.

	Democrats	Republican
Democrats	54	0
Republican	3	30

TABLE 13. House votes Winnow2 results.

	Democrats	Republican
Democrats	50	0
Republican	7	30

TABLE 14. House votes Naive Bayes results.

The Winnow2 algorithm outperforms Naive Bayes slightly, which is logical, as this is a set boolean inputs, and so Naive Bayes has no advantage here.

5. Discussion of Algorithm Behavior

The results above are intuitive. Winnow2 tends to excel in situations where the inputs are all boolean, whereas Naive Bayes has an advantage with discrete multivalued and continuous inputs. This seems like a good rule of thumb in deciding between the two algorithms. Otherwise, their behavior is quite similar in terms of accuracy for a given problem, as they search similar representation spaces (univariate and linear).

One big boost given to the Winnow2 algorithm in this paper is the manual discretization of continuous variables along favorable boundaries. One could surmise that given a significanly larger feature space, this would become quite cumbersome, and may not be possible.

6. Summary

This paper introduced two learning algorithms: Winnow2 and Naive Bayes. Winnow2 uses weights and a linear threshold to learn the boundary between two classes. It can only be applied to boolean inputs and outputs. Naive Bayes uses a conditional independence assumption to learn the likelihood of an instance belonging to a class. It can be used on multivalued discrete and continuous inputs. Its generally applied to classification problems, but could likely be adapted to regression problems.

This paper then applied both algorithms to a set of classification problems. It found that the two algorithms perform similarly, with Winnow2 excelling with large numbers of boolean inputs, and Naive Bayes exceling with continuous inputs.

References

[1] Keming Yu David J. Hand. Idiot's bayes - not so stupid after all. *International Statistical Review*, 63:385, 2001.
[2] R.A Fisher. Iris. URL: `https://archive.ics.uci.edu/ml/datasets/Iris`.
[3] B. German. Glass identification data set. URL: `https://archive.ics.uci.edu/ml/datasets/Glass+Identification`.
[4] Nick Littlestone. Learning quickly when irrelevant attributed abound: A new linear-threshold algorithm. *Machine Learning 2*, pages 285–318, 1988. URL: `https://link.springer.com/content/pdf/10.1023/A:1022869011914.pdf`.
[5] R.S.n Michalski. Soybean (small) data set. URL: `https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29`.
[6] Jeff Schlimmer. Congressional voting records data set. URL: `https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records`.
[7] Dr. WIlliam H. Wolberg. Breast cancer wisconsin (original) data set. URL: `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29`.