Winnow-2 vs Naïve Bayes (CS605.449 – Project 1)

By Max Robinson

Abstract

In this experiment we compared the performance of the Winnow-2 algorithm against that of the Naïve Bayes classifier algorithm on 5 different data sets. The data sets were varied in their original representation and were thus formatted into a standardized format for both algorithms. The result of the experiment showed the that Naïve Bayes classifier performed slightly better than the winnow-2 algorithm across all data sets, having a smaller error rate on 10 out of the 17 classification experiments run and overall usually performed better on 4 out of the 5 data sets.

Problem

The problem that we are investigating in this experiment is which algorithm Winnow-2 or Naïve Bayes performs better on the following data sets: Wisconsin Breast Cancer, Glass Identification, Iris variety, Soybean Variety, and votes taken in the house in 1984. For this particular experiment I hypothesize that the Naïve Bayes classifier will have a lower overall error rate across the different datasets than the winnow-2 algorithm.
The Winnow-2 algorithm is a fairly simplistic algorithm that works only with binary features. Since most of the data that we will be using does not originally have binary features, and only has them after being transformed it is likely that the transformation will hurt the performance of Winnow-2.
The Naïve Bayes classifier will use the same data but since the data should hold its distributions through being transformed into binary features, I believe that it will perform better than the Winnow-2 algorithm.

Algorithm Implementation

Winnow-2

The winnow-2 algorithm is an algorithm that uses binary features as input and a binary class to learn a set of weights that are applied to each feature when making a prediction. There is a weight associated with each feature in the feature vector that the model is learning over. For example, if the data set has features of length 5, excluding the class value in a training feature, there will be 5 weights i n the Winnow-2 model.

To make a prediction the Winnow-2 algorithm uses the linear combination $f(x) = \sum w_i x_i$ where x is a feature vector x_i is a feature in the feature vector and w_i is the associated weight in the Winnow model. If the result is above a threshold θ then it is part of the positive class, otherwise it is not and is part of the negative class.

When learning, the weights are adjusted through promotion and demotion of values given if the algorithm made a correct prediction or not. Correct predictions incur now change to the weights. If the classification is a false positive a demotion takes place. If the classification is a false negative, a promotion takes place. When adjusting weights, only the weights that were applied to a non-zero feature are adjusted.

Naïve Bayes Classifier

The Naïve Bayes Classifier works quite differently from the winnow-2 algorithm. The Naïve Bayes classifier works by essentially a "database" of records, from which the distribution of the data is reduced. Specifically, the values calculated are $P(f|c)$ for some feature value for a given feature with respect to some classification label. In doing this, we are assuming that each feature value is independent of every other feature value, given the class. Irina Rish, in per paper talks about some of the implications of this.

For this experiment, we transformed all of the data into binary feature sets so each feature is referenced by feature number (i.e. feature 0,1,2,3 etc). The given values that are possible for any given feature number or $f = 0 \; or \; 1$, since we are using binary features. Since we are also using binary classes, positive or negative class, the possible values for classes are $c = 0 \; or \; 1$. As such the possible probabilities found for a given feature are $P(0|0), P(0|1), P(1|0), P(1|1)$.

Each of these probabilities are calculated from the training set. When a new feature vector needs to be classified, the probabilities for each feature value are looked up for each class, and multiplied together. This value is then normalized for each class based on the probability of that class in the entire dataset. The resulting classification is the class that has the highest probability for that class.

For this specific implementation, we also used a +1 smoothing tactic to ensure that no probabilities for a given feature were zero. If any probabilities were zero, this can cause the entire probability to be zero, which would provide often incorrect predictions, or no prediction at all for a class.

Experimental Approach

The approach for this experiment was 2 fold. The first part was done through data preprocessing, and the second part was done through how the experiments were run. The metric used for comparison of the two algorithms in this experiment was error rate, which is simply the number of incorrect predictions divided by the total number of items in the test set.

Data Pre-processing

Each data set was pre-processed to turn all of the features into binary features, 0's and 1's. The process for each data set was a little different, since they all had different ranges of values and types of values.

For the Breast Cancer data set, the ID feature was removed from the set since it was a unique id and does not provide correlation with the data. The remaining features all have values between 1 and 10 so nine new features were created for each one and a one's hot encoding was applied to each new feature. As a result, there were 90 total features, plus the class label per feature vector.

For the Glass data set, the unique id was removed. All values were floats, and had a max and min range for the data set. These values were then binned into 6 bins. A one's hot encoding was applied to which bin a given value fell into.

The iris dataset was also real numbers based and had a maximum and minim value for each feature. The same tactic of binning and one's hot encoding was used for this dataset. 6 bins were also used.

The soybean dataset was integer based, with a maximum value in the entire data set of 6. As such, each integer feature was given 6 binary features, and a one's hot encoding was applied. For missing f eatures, a random value between 0 and six was generated and used in its place.

The house votes dataset was binary based, so all the values were transposed into 1's and 0's. For missing values, a random number 0 or 1 was generated for that place. This data set had the largest portion of missing data.

Classification

To get comparable results, each algorithm was run on each data set. In addition, each data set was tested with each possible class as the positive class. For data sets with more than 2 classes, this means that the positive class used was selected at run time, and 1 class was selected. All other records that were or a different class were then lumped into the negative class. Each possible class value for each dataset was used as the positive class. This allows for any sort of odd balance of the classes in the data to possibly be noticed.

For each exercise, an error rate was calculated for the test set. That error rate is what is compared and displayed in the results section.

Results

After the running the experiment as described in the above sections, the error rates for each dataset and each positive class used were calculated. The results can be seen in table 1.

Error Rate Table	Positive Class Used	Winnow-2	Naïve Bayes
Breast Cancer	Benign	0.0815450643776824	0.02145922746781116
	Malignant	0.0815450643776824	0.04291845493562232
Glass	1	0.3611111111111111	0.3055555555555556
	2	0.3472222222222222	0.3472222222222222
	3	0.19444444444444445	0.4444444444444444
	5	0.09722222222222222	0.16666666666666666
	6	0.09722222222222222	0.2222222222222222
	7	0.125	0.1111111111111111
Iris	Iris Setosa	0.0	0.0
	Iris Versicolour	0.12	0.14
	Iris Virginica	0.12	0.1
Soybean (small)	D1	0.0	0.0
	D2	0.29411764705882354	0.0
	D3	0.058823529411764705	0.0
	D4	0.0	0.058823529411764705
House-Votes	Democrat	0.18620689655172415	0.06896551724137931
	Republican	0.06896551724137931	0.07586206896551724

Table 1. Error Rate of Winnow-2 and Naïve Bayes per Dataset and Positive Class

Behavior

The behavior of the algorithms was a little surprising. For many of the data sets, the two algorithms were relatively close, within a few percentage points of each other up to about 6%. On other datasets, and certain positive classes there is a wide gap in performance. One example is Glass with positive class 3. The Naïve Bayes algorithm is more than 20% worse than the Winnow-2 algorithm. In a different case, soybeans positive class D2, winnow-2 is almost 30% worse than Naïve Bayes.

Overall, the algorithms performed fairly closely to each other. Naïve Bayes classifier did perform slightly better than the winnow-2 algorithm across all data sets. It had a smaller error rate on 10 out of the 17 classification experiments run. However, as mentioned previously, there are some outliers.

The dataset where the Naïve Bayes classifier did the worst overall was on the Glass dataset. It performed worst or approximately the same as winnow-2 on all of the positive classes used. This seemed a little odd since on each of the other datasets Naïve Bayes did about as well if not better.

I believe that this performance difference is a result of how the data was translated from the real values in the original glass dataset and the binned version. Unlike the iris dataset, where the ranges for each feature value were roughly the same, the glass features were not so. Many had larger ranges than others and some had smaller ranges than others. As a result, some parts of the data might not have been well captured in the binary feature sets since many of the values could have been lumped under the same feature in the one's hot encoding.

I think the winnow-2 algorithm performed better in this scenario because it does not rely exactly on the training data that was handed to it in such a direct manner. In addition, in Nick Littlestone's paper about using Winnow-2 with noisy attributes, he finds that Winnow-2 is actually reasonably good at filtering out noisy attributes or attributes that don't matter. I believe this gave Winnow-2 a little bit of an edge in this dataset where the attributes might not have been as well distributed as possible and caused some noisy attributes for the Naïve Bayes implementation.

Summary

Overall, both algorithms performed quite well on the data sets provided, disregarding a few outliers. On many of the dataset and exercises the error rate, which was used to compare th e two algorithms, was below 10%. The glass dataset seemed to be a tricky and troublesome dataset for both the Naïve Bayes classifier. Each algorithm performed worse than usual, above or at the 10% error rate on each positive class. Outside of this, the Naïve Bayes Classifier slightly outperformed the Winnow-2 algorithm by performing better or equal to Winnow-2 across 4 of the 5 datasets.

References

Algorithms

Winnow-2: Redundant Noisy Attributes, Attribute Errors, and Linear-Threshold Learning Using Winnow by Nick Littlestone

Naïve Bayes: An empirical Study of the Naïve Bayes Classifier by Irina Rish
https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf
(Note: this reference was also used in understanding Naïve Bayes in order to implement it).

Data Sources

Breast Cancer— https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29
Glass — https://archive.ics .uci.edu/ml/datasets/Glass+Identification
Iris — https://archive.ics.uci.edu/ml/datasets/Iris
Soybean (small) — https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29
Vote — https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records