JMLR: Workshop and Conference Proceedings 1:1–4, 2016 Submitted 9/11/2016; Published 9/11/2016

Introduction to Machine Learning - EN 605.449.81 - Lab 1

Max Henry mhenry22@jhu.edu

Abstract

// TODO

Keywords: Winnow-2, naive Bayes

1. Problem Statement

The purpose of the assignment was to implement the Winnow-2 and Naive Bayes algorithms and test their performance on five datasets from the UCI Machine Learning Repository. These datasets include the University of Wisconsin's breast cancer database, the glass identification dataset, the well-known iris dataset, the soybean dataset, and the U.S. House of Representatives dataset.
Because the datasets a range of discrete, continuous, and multi-class problems, pre-process was needed. Considering the widespread success of naive Bayes on multiple problem domain, I hypothesize that it will beat Winnow-2's performance for every dataset.

2. Algorithms

2.1. Winnow-2

Winnow-2 is a linear classifier similar to the Perceptron. The classifier learns weights for each attribute and, when asked to make a prediction, uses these weights to sum up a value that is inputted into a threshold function. The output of this threshold function becomes the classifier's prediction for the unseen instance. If the prediction is correct, the weights are left alone. Otherwise, the weights are updated based on how the prediction was incorrect. If the prediction was 0 and the true class of the unseen instance is 1, the weights undergo a process called "promotion". If the prediction was 1 and the true class of the unseen instance is 0, the weights undergo a process called "demotion".

2.1.1. Model

The Winnow-2 algorithm learns a classifier of the form $f(x) = \sum_{i=1}^{d} w_i x_i$ where d is the number of attributes in an instance, w_i is the weight associated to an attribute and x_i is the attribute.

2.1.2. Threshold

The output of $f(x)$ is inputted into $h(x)$ to generate a prediction for the unseen instance. $h(x)$ returns 1 if $f(x) > \theta$ and 0 otherwise. θ is a threshold chosen by the practitioner.

2.2. Promotion

When the classifier incorrectly guesses 0 (in a two-class problem), the weights are updated. If the unseen instance's x_i is 1 then w_i becomes αw_i, otherwise the weight is left alone.

2.3. Demotion

When the classifier incorrectly guesses 1 (in a two-class problem), the weights are updated. If the unseen instance's x_i is 1 then w_i becomes w_i/α, otherwise the weight is left alone.

2.4. Naive Bayes

Naive Bayes predicts a class label from $argmax_c P(c|f_1, ..., f_d)$ where c is the certain class label and $f_1, ..., f_d$ are the attributes of the unseen instance. It leverages Bayes Rule by trying to predict a class label given the unseen instance. It makes the assumption that all features f_i are conditionally independent of each other given the class label. This means the calculation becomes $argmax_c P(c)\Pi_{i=1}^{d} P(f_i|d)$.

3. Experimental Approach

3.1. Data Cleaning

Due to the unprocessed nature of the data, each continuous-valued dataset was discretized using 10 bins of equal size between the attributes min and max value. In order to be fair to both algorithms, discretized attributes were translated with a one-hot encoder, mostly to 10-bit strings. In the case of multi-class (N classes), the original dataset was split into N datasets, one for each class, and translated into a 0/1 class setup. In addition, any missing values were replaced with a random number draw from the distribution of instances with the same class label.

3.2. Experiments

Each algorithm was run against each of the five datasets using 10-fold cross-validation. The success rate, calculated as the number of correctly labeled instances out of the total number of instances in the test set, was averaged over every fold. This number was used to compare the algorithms. Note that this numbers were not tested for statistical significance.

4. Results

Dataset	Winnow-2	Naive Bayes
Breast Cancer	95.13%	96.42%
Glass (building_windows_float_processed)	65.00%	71.45%
Glass (building_windows_non_float_processed)	67.27%	68.18%
Glass (vehicle_windows_float_processed)	91.58%	86.04%
Glass (vehicle_windows_non_float_processed)	100.00%	100.00%
Glass (containers)	93.51%	95.82%
Glass (tableware)	90.65%	95.84%
Glass (headlamps)	94.37%	96.73%
Iris (Setosa)	100.00%	100.00%
Iris (Versicolour)	85.33%	95.33%
Iris (Virginica)	94.00%	91.33%
Soybean (D1)	100.00%	100.00%
Soybean (D2)	100.00%	100.00%
Soybean (D3)	54.50%	100.00%
Soybean (D4)	100.00%	100.00%
Voting	95.62%	91.97%

Table 1: Success rates of both algorithms on various datasets

5. Conclusions

Naive Bayes outperformed Winnow-2 on eight datasets and tied it on five. This disproves my hypothesis that naive Bayes would outperform Winnow-2 on every dataset. In the three cases where Winnow-2 wins, the gap is less than five percentage points. Performances were not judged for statistical significance.

Given accuracy rates of 80%+ for most cases, suggests the most concepts learned in each dataset were linearly separable. It is interesting to point out that the three cases where Winnow-2 performance dropped below 70%, the naive Bayes classifier outperformed it. This may suggest that those three datasets have concepts that are not linearly separable indicating a case where the linear classifier naive Bayes can still be applied to non-linearly separable data.

6. Summary

In this lab, I implemented the Winnow-2 and naive Bayes algorithms. Their performance was compared on five datasets (split by class in the multi-class case). In general, naive Bayes seemed to perform better but results were not tested for statistical significance.

7. References

Winnow (algorithm). Retrieved September 10, 2016, from en.wikipedia.orgwikiWinnow_(algorithm)

Nick Littlestone (1988). "Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm", Machine Learning 285318(2)

Naive Bayes classifier. (n.d.). Retrieved September 10, 2016, from en.wikipedia.orgwikiNaive_Bayes_classifier