CSCI447 - Assignment 1

Chris Major CHRISMICHELMAJOR@HOTMAIL.COM
Farshina Nazrul-Shimim FARSHINA287LEO@GMAIL.COM
Tysen Radovich RADOVICH.TYSEN@GMAIL.COM
Allen Simpson ALLEN.SIMPSON@MOTIONENCODING.COM

Editor:

Abstract

In the interests of experimentation regarding an entry-level area of study within machine learning, the Naive-Bayes Algorithm was used to analyze five sets of data and demonstrate acceptable levels of accuracy in classification. Under the framework of supervised learning, a series of tests involving data pre-processing, training, classification, and result validation were constructed and written in code. The data intended for processing, analysis, and classification was provided by the UCI Machine Learning Repository and divided by our team into training and testing subsets in the series of experiments using 10-Fold cross-validation techniques. Five implementations of the Naive-Bayes Algorithm were written in F# and tested on each data set, with the results analyzed and discussed in the context of classification error.

1. Introduction

The Naive-Bayes Algorithm is regarded as a point of entry for students in machine learning due to its relatively simple premise, ease of implementation in code, and versatility within conditionally independent data sets. As with all machine learning algorithms, however, its inductive methods do not guarantee perfect classification or prediction for all possible observations. The only guarantee for an operational implementation of the Naive-Bayes Algorithm is the presentation of a probability in the success of the classification.

To demonstrate these principles, five experiments were conducted with an implementation of the Naive-Bayes Algorithm in the F# programming language, a language used by a wide range of machine learning researchers and acclaimed for its brevity in implementing functions. The data sets chosen for these experiments can be found within the UCI Machine Learning Repository(Dua and Graff (2017)).

2. Problem Statement

WIthin each experiment, two cases are considered for the training and classification of each data set: the original data arrangement, and an arrangement where 10% of the features are randomized. Depending on the data set, denoted by the file type `.data` and the file notes associated with each data set, denoted by the file type `.names`, pre-processing on the data may be required to account for missing values, various data types, and the elimination of unnecessary attributes.

Generally, we hypothesize that data sets with more complexity in their attributes are more likely to experience larger error in results after the shuffle than data sets with less features, especially if there are randomly distributed values across certain attributes or large numbers of attributes. More specifically, we hypothesize that randomizing the data in any of these given data sets will lead to significantly higher loss values.

The specific data sets used in the experiments are as follows.

2.1 Breast Cancer Data Set

The file `breast-cancer-wisconsin.data` consists of 699 various cell samples from clinical cases, provided by the University of Wisconsin. The goal was to evaluate whether a certain sample can be classified as benign or malignant, given the properties of various samples.

Each attribute, save the assigned ID value of each observation, has an integer value ranging from 1 to 10. There are no missing values in the data set.

2.2 Glass Data Set

The file `glass.data` consists of 214 various glass samples from forensic investigations. The goal was to evaluate the correct type of glass given certain properties for a sample, using the aforementioned algorithm.

There are seven possible classes for the data set and nine attributes. Each data point is a floating-point value with a different range per feature. There are no missing values in the data set.

2.3 Iris Data Set

The file `iris.data` consists of 150 various plant samples from an iris flower database. The goal was to evaluate which of three possible iris varieties a sample best fit.

Historically, `iris.data` has proven to be a challenging data set for analysis. There are three possible classes - with 1 being linearly separable from the other 2 linearly-dependent classes - and only four listed attributes. Due to these dependencies, the Naive-Bayes Algorithm should not be expected to perform well. There are no missing values in the data set.

2.4 Soybean Data Set

The file `soybean-small.data` is a subset of `soybean-large.data`, consisting of 47 different soybean instances. The goal was to evaluate which of four possible distinctions a soybean sample best fit.

There are two soybean data sets, consisting of 35 different, integer-valued attributes. As many of these attributes offer little to no valuable information for the sake of this experiment, select attributes were removed from the analysis. There are 4 possible classes for classification and no missing values in the data set.

2.5 House Votes Data Set

The file `house-votes-84.data` consists of 435 instances of voting records cast in a 1984 Congressional meeting. The goal was to determine which political party with which a voting record best aligned.

At a glance, the values for each feature are boolean, but there are multiple instances of undefined values that correspond to "present" votes - neither affirmative nor negative. These were treated prior to experimentation to reflect this change. There are sixteen attributes and 2 classes.

3. Experiments

To test the effectiveness of the Naive-Bayes Algorithm on conditionally-independent data sets, code was written in F# to parse each data file into its corresponding attributes, perform the analysis, and calculate the 0/1 Loss of the results to assure functionality.

A function called `trainingDataSet` was written to parse the file line by line and assign each value to an array of attribute values. The observation ID values were recorded, despite remaining unused in the rest of the experiments. The values representing classes were interpreted in terms of a Class object, for the sake of easy analysis.

The next function, called `classify`, implements the equations needed for Naive-Bayes. The classification function for any observation x is as follows:

$$class(x) = argmax(Q(C = c_i) \times \sum_{Y_j=1}^{d} F(A_j = a_k; C = c_i))$$

Here, `x` represents the observation, A_j and A_k represent given attributes, C and C_i represent given classes in the integration, and $Q(c)$ and $F(a, c)$ are as follows:

$$Q(C = c_i) = \frac{\#\{x \in c_i\}}{N}$$

$$F(A_j = a_k; C = c_i) = \frac{\#\{(x_{A_j} = a_k) \wedge (x \in c_i) + 1\}}{N_{c_i} + d}$$

These equations are an algebraically rewritten form of Bayes' Rule (described in Mitchell (2015)), which forms the basis of the Naive-Bayes Algorithm. The $\#\{pred\}$ function returns the number of times that the predicate given by pred is matched in the argument.

Our validation error is calculated using 0/1 Loss functions, based on the number of elements in the sets and tuples of the classes. This method helps to determine the performance of the algorithm and the validation error.

The evaluation between training and testing sets is conducted through 10-Fold Cross Validation, as described by Dietterich (1998). The data set is divided into 10 equally-sized subsets, where each set is used for training as the others are tested. The results are highlighted and discussed in Results and Analysis.

These functions are repeated for the original data set and the shuffled data set, in which 10% of the features are randomly shuffled.

This experiment code was written for generic cases in F#, with several scripts written for testing and assessing individual data sets.

4. Results and Analysis

4.1 Breast Cancer Data Set

The experiment was performed on `breast-cancer-wisconsin.data`, with the terminal reporting the 0/1 Loss for each of the K-Folds calculated. The results of the experiment are listed below in Table 1.

Data Set	Average 10-Fold Cross Validation Error
Full	2.4604% ±0.3%
Shuffled 10% Features	2.5473% ±0.5%

Table 1: Average % of failures for all `breast-cancer-wisconsin.data` validation sets

The largest error in this run corresponds to the shuffled data, though compared to the rest of the tested data sets, this experiment has proven to have the lowest classification error. The simplicity of the data and the conditionally independent attributes and classes are to be credited for these results, therefore our hypothesis is refuted by this specific data set.

4.2 Glass Data Set

Next, the experiment was performed on `glass.data`, with the terminal reporting the 0/1 Loss for each of the K-Folds calculated. The results of the experiment are listed below in Table 2.

Data Set	Average 10-Fold Cross Validation Error
Full	35.1340% ±2%
Shuffled 10% Features	36.2554% ±2%

Table 2: Average % of failures for all `glass.data` validation sets

`glass.data` proved to be the most problematic experiment as the attributes were listed with floating point values, requiring additional treatment to ensure proper predicate matching. As such, the robustness of the algorithm is called into question, contributing to the high error rates shown in the table. The shuffling of the data only further amplifies the error, contributing to the increased rate. Therefore, we conclude the null hypothesis is refuted by this data set.

4.3 Iris Data Set

Next, the experiment was performed on `iris.data`, with the terminal reporting the 0/1 Loss for each of the K-Folds calculated. The results of the experiment are listed below in Table 3.

Despite the note of caution from the lecture regarding the conditional dependencies of `iris.data`'s classes, the algorithm proved to be acceptable in testing and classification. Due to the low number of attributes amidst a high number of observations, the shuffling significantly affected the error rates in classification. Therefore, we can safely conclude that the null hypothesis is refuted by this data set.

Data Set	Average 10-Fold Cross Validation Error
Full	7.2667%
Shuffled 10% Features	10.9200%

Table 3: Average % of failures for all `iris.data` validation sets

4.4 Soybean Data Set

Next, the experiment was performed on `soybean-small.data`, with the terminal reporting the 0/1 Loss for each of the K-Folds calculated. These results are listed below in Table 3.

Data Set	Average 10-Fold Cross Validation Error	Attributes Omitted
Full	20.7250% ±0.3%	None
Shuffled 10% Features	32.1350% ±0.3%	None
Full	2.6900% ±0.3%	Singleton Values
Shuffled 10% Features	36.0800% ±0.4%	Singleton Values

Table 4: Average % of failures for all `soybean-small.data` validation sets

Interestingly, `soybean-small.data` reported the best error rate for an un-shuffled trial with no singleton values; however, the presence of the singleton values greatly influences the validation error. Again, the shuffled values demonstrate a greater error rate, thus the null hypothesis is refuted for this experiment. Additionally, the presence of singleton attributes are as detrimental to the data as shuffling the data.

4.5 House Vote Data Set

Data Set	Average 10-Fold Cross Validation Error
Full	10.0666% ±0.005%
Shuffled 10% Features	10.2743% ±0.06%

Table 5: Average % of failures for all `house-votes-84.data` validation sets

The validation errors of the full and shuffled vote data sets do not vary significantly. Though the attributes are not strictly boolean, as there are neutral votes in the data, the range of possible values in this data set is relatively simple and therefore reduces the difference in error between the cases. While our hypothesis technically holds for this experiment trial, the difference of 0.07% suggests no significant change in validation and therefore it is safer to conclude that the null hypothesis is refuted by this case.

5. Conclusion

In conclusion, it has been shown that the performance of the Naive-Bayes Algorithm on this series of practice data sets falls within expectations. The validation error is tied to the complexity of the data provided and, more dramatically, to the shuffling of features within a data set. We therefore conclude that our hypothesis is affirmed by these experiments.

Discussion regarding the limitations of the Naive-Bayes Algorithm spawns from this point as the various conditions reflected across the five different experiment trials reflect countless machine learning classification problems. These points prompt the need for appropriately chosen adjustments to the method or the consideration of other methods entirely. Regardless, the algorithm has proven, per the experiments performed, to be an effective method for basic classification of conditionally independent data sets.

References

T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, Oct 1998. doi: 10.1162/089976698300017197.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Tom M. Mitchell. *Machine Learning*. McGraw Hill, 2015.