Journal of Machine Learning Research 1 (2020) 1-7 Submitted 9/20

Winnow-2 vs Naive Bayes Algorithm

Jieyi Chen

September 12, 2020

1. Introduction:

In this paper, we compare the performance of two supervised learning models (Winnow-2 and Naïve Bayes Algorithm) in classifications problems. The classification model is trying to predict a class label based on the attributes that are associated with the observation. Binary classification and multi-class classification are the two common types of classification problems. The ability to correctly classify a dataset is integral to many fields of the society, therefore, this area of the machine learning concept has gained lots of interests in recent years. This paper introduces two common algorithms and discusses their performances in different types of datasets. Winnow-2 is a machine learning model that builds a linear classifier from the labeled data and is applicable to multi-dimensional data [1]. Naïve Bayes Algorithm is a classification technique that applies the concept of Bayes theorem with the strong assumption that the features values are conditionally independent given the class [2]. We expect the Naïve Bayes Algorithm performs better, especially on datasets with more than 2 classes because it is more generalized. For the Winnow-2 Algorithm, we need to create one classifier for each class, which can lower its accuracy. In this project, we use five datasets from the UCI repository [3] and compare these two algorithms' classification accuracy to test this hypothesis.

The rest of the paper is organized as follows. Section 2 describes the algorithms that are implemented. Section 3 discusses the experimental approach. Section 4 presents the results of the experiments. Section 5 discusses the results. Section 6 concludes.

2. Algorithm Implementation:

2.1 Winnow-2 Algorithm:

Winnow-2 Algorithm is a supervised learning algorithm developed by Nick Littlestone [2]. This online learning algorithm receives data instances with two classes. The classes are binary values, 0 and 1. 0 means that the instance does not belong to the class and 1 means it belongs to the class. The algorithm makes predictions for the observations by assigning 0 and 1. If the learner assigns 0 to the data instance, it means that the learner predicts the instance does not belong to the class. If the learner assigns 1 to the observation, it means that the learner believes it does. The algorithm would receive feedback on whether its prediction is correct. If it makes correct prediction, nothing happens. If it does not, the model will learn by doing promotion and demotion based on its prediction, which is listed in Table 1. If the learner predicts the observation belongs to class 0, but in reality it belongs to 1, then it does promotion. On the contrary, if the learner's prediction is class 1 but the true label is 0, it does demotion.

Learner Process		
Learner's Prediction	**True Label**	**Process**
0	1	Promotion
1	0	Demotion

Table 1 - Learner Process

Model:

Given a dataset with $x_1 \ldots x_n$ observations, Winnow-2 algorithm creates a linear classifier using the steps below.

1. Initialize all the weights for the attributes to 1
2. Calculate the weighted sum using the weights and attributes

$$f(x) = \sum_{i=1}^{d} w_i x_i$$

- d = total number of attributes
- w_i = weights for the attributes
- x_i = attributes

3. Set the threshold θ
4. If the weighted sum is greater than the threshold, the learner assigns 1 to the data instance. Otherwise, it assigns 0.

$$h(x) = \begin{cases} 1 & f(x) > \theta \\ 0 & otherwise \end{cases}$$

5. If the learner incorrectly predicts the class, it will learn by doing promotion and demotion.

The promoting process is as follows: if $x_i = 1$, the weight for i_{th} attribute will be updated by multiplying α. Otherwise, the weight remains the same. The demoting process is as follows: if $x_i = 1$, the weight for i_{th} attributes will be updated by dividing α. Otherwise, the weight remains the same.

Promotion:

$$w_i = \begin{cases} aw_i & if\ x_i = 1 \\ w_i & if\ x_i = 0 \end{cases}$$

Demotion:

$$w_i = \begin{cases} \frac{w_i}{\alpha} & if\ x_i = 1 \\ w_i & if\ x_i = 0 \end{cases}$$

2.2 Naïve Bayes Algorithm:

Naïve Bayes Algorithm is a supervised machine learning algorithm that applies the concepts of Naïve Bayes theorem and posterior probability.

Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Posterior Probability:

$$\text{Posterior} \propto \text{Prior x Likelihood}$$

$$P(C|A) \propto P(C)P(A|C)$$

where C = class and A = Attributes

Posterior: The probability of the observation in a particular class given the attributes
Prior: The probability of the observation in a particular class
Likelihood: The probability of the observation having the attributes given the class information

Model:

Training Phase:

- Calculates the probability of each class
- Uses the formulas in Table 2 to calculate the conditional probability for each attribute given the class information

Testing Phase:

- Uses the formulas in Table 3 to calculate the conditional probability for each class given the attribute information
- The class with the highest conditional probability is the prediction returned from the model

Training Phase (Binary)			
		Class (C_i)	
		$C_i = 0$	$C_i = 1$
Attribute (A_i)	$A_i = 0$	$P(A_i = 0 \mid C_i = 0) = \frac{P(A_i=0 \cap C_i = 0)}{P(C_i = 0)}$	$P(A_i = 0 \mid C_i = 1) = \frac{P(A_i=0 \cap C_i = 1)}{P(C_i = 1)}$
	$A_i = 1$	$P(A_i = 1 \mid C_i = 0) = \frac{P(A_i=1 \cap C_i = 0)}{P(C_i = 0)}$	$P(A_i = 1 \mid C_i = 1) = \frac{P(A_i=1 \cap C_i = 1)}{P(C_i = 1)}$

Table 2 - Conditional probability calculated in the training phase

Testing Phase (Binary)	
Class (C_i)	Probability $P(C_i \mid A_i)$
$C_i = 0$	$(\sum_{i=1}^{n} P(A_i = 0 \mid C_i = 0)) * P(C_i = 0)$
$C_i = 1$	$(\sum_{i=1}^{n} P(A_i = 0 \mid C_i = 1)) * P(C_i = 1)$

Table 3 - Conditional probability calculated in the testing phase

Datasets:

The datasets are from UCI Machine learning Repository [1]. Table 4 summarizes the properties of the five datasets.

Dataset Property Summary Table					
Hyperparameter	**Breast Cancer Dataset**	**Glass Dataset**	**Iris Dataset**	**Soybean Dataset**	**Vote Dataset**
# Attributes	9	9	4	35	16
# Examples	699	214	150	47	435
Class	Binary	Multiclass	Multiclass	Multiclass	Binary

Table 4 - Dataset Property Summary Table

Data Preprocessing Steps:

The breast cancer and vote datasets have missing values and we use mean imputation and the conditional probability of the attribute given the class to handle them. We separate the observations with missing values based on their classes and replace those values with the average of the attribute in the class. In addition, many of the attributes in the datasets are multi-value discrete or real-valued. Therefore, we need to use discretization to transform the numerical variables into categorical. In this case, we apply the equal-width binning method to evenly divide the multi-value discrete variables into three or two bins, depending on the number of attributes in the dataset. In addition, we use the one-hot encoding method to create a separate Boolean attribute for each value so that Winnow-2 algorithm can be executed with the transformed variables.

Binning Algorithm:

1. Find the maximum and minimum value of the attribute
2. The bin size is the differences between the maximum and minimum value divided by the number of bins
3. Create a dictionary to store the bin number and its corresponding interval
4. Check each value in the attribute column and assign it to the corresponding bin
5. Repeat the steps above for all the multi-value discrete variables

One-Hot Encoding Algorithm:

1. For each value in the attribute columns
    a. Create a list of zeros with its length equals to the number of bins
    b. Use the bin number to identify the position in the list created in Step 1 and change that value from 0 to 1
2. Repeat the steps above for all the multi-value discrete variables

Hyperparameter Tuning Process:

In order to find the optimal hyperparameters, we reserve 10% of the data for tuning, and 2/3 of the remaining 90% for training and 1/3 for testing. The hyperparameters for the Winnow-2 algorithm is θ and α and the ones for the Naïve Bayes Algorithm are p and m. Table 6 and 7 in the result section summarizes the accuracy for each of the datasets using the corresponding optimal hyperparameters.

Winnow-2 Algorithm Implementation:

Winnow-2 Algorithm for Binary Classification Dataset:

1. Train the model with the training set using the procedure above
2. Apply the weights trained from the training set to the tuning set. If the weighted sum is greater than the threshold, the learner's prediction is 1. Otherwise, it is 0.
3. Create a dataframe to store the accuracy of the model on the tuning set and its corresponding hyperparameters and weights vectors
4. Apply the optimal hyperparameters found on step 3 to the test set

Winnow-2 Algorithm for Multi-class Dataset:

1. Create one classifier for each class in the dataset
    - For each class, we assume the observations in the class as one category and the data instances in any other classes as another group
    - We train the model for each class and get the weights vector, which results in K weights vectors assuming we have K classes
2. Apply the K weights vectors to the tuning set, for each observation in the tuning set, we have K weighted sum. The observation is classified as the class with the highest weighted sum.
3. Create a dataframe to store the accuracy of the model on the tuning set and its corresponding hyperparameters and weights vectors
4. Apply the optimal hyperparameters found on step 3 to the test set

Naïve Bayes Algorithm Implementation:

1) Create a table for storing the conditional probability for each attribute given the class information
2) Iterate through each observation in the training dataset and store the column name and value in a dictionary
3) Use the dictionary created in step 2 to search for the probability in the probability table created in step 1 and save these conditional probabilities in a list
4) Multiply all the conditional probability in the list and the class probability, which gives the conditional probability of class given the attribute information. The class with the highest conditional probability is the prediction from the model

4. Result:

Algorithm Accuracy Summary Table					
Algorithm	**Breast Cancer Dataset (Binary)**	**Glass Dataset (Multiclass)**	**Iris Dataset (Multiclass)**	**Soybean Dataset (Multiclass)**	**Vote Dataset (Binary)**
Winnow 2	0.96	0.48	0.93	0.93	0.98
Naïve Bayes	0.95	0.7	1	0.93	0.94

Table 5 - Algorithm Accuracy Summary Table

Hyperparameters Summary Table					
Hyperparameter	**Breast Cancer Dataset (Binary)**	**Glass Dataset (Multiclass)**	**Iris Dataset (Multiclass)**	**Soybean Dataset (Multiclass)**	**Vote Dataset (Binary)**
θ	0.5	0.1	0.1	0.25	0.1
α	4	2	3	3	2
Accuracy	0.96	0.48	0.93	0.93	0.98

Table 6 - Hyperparameters Summary Table

Hyperparameters Summary Table					
Hyperparameter	**Breast Cancer Dataset (Binary)**	**Glass Dataset (Multiclass)**	**Iris Dataset (Multiclass)**	**Soybean Dataset (Multiclass)**	**Vote Dataset (Binary)**
p	0.6	0.001	0.001	0.001	0.001
m	1	1	1	1	1
Accuracy	0.95	0.7	1	0.93	0.94

Table 7 - Hyperparameters Summary Table

5. Discussion:

As we can see from Table 5, Winnow-2 Algorithm did a better job in classifying the datasets with two classes while the Naïve Bayes Algorithm performs better in the multi-class datasets. Both algorithms did not do an excellent job in classification for the glass dataset, but they outperformed in the rest of the datasets. Table 6 shows that $\theta = \{0.1, 0.25, 0.5\}$ and $\alpha = \{2, 3, 4\}$ are the optimal hyperparameters for the Winnow-2 Algorithm. Table 7 shows that $p = \{0.001, 0.6\}$ and $m = \{1, 10\}$ are the optimal hyperparameters for Naïve Bayes Algorithm. These results have shown that part of our hypothesis is correct, but it does not take into the fact that Winnow-2 Algorithm performs better in binary classification.

6. Conclusions:

This project helps us to gain a better understanding of the Winnow-2 and Naïve Bayes Algorithms. In addition, we have implemented the binning and one-hot encoding from scratch to ensure that the multi-value discrete variables can be used in algorithms similar to Winnow-2. Moreover, we have learned how to tune the hyperparameters to achieve a higher accuracy in the test set. This paper can help analysts to decide which model to implement when they encounter the classification problems. Winnow-2 algorithm will be suitable for binary classification while Naïve Bayes algorithm performs better with the multi-class datasets. Some future work on how to optimize Winnow-2 algorithm for multi-class datasets can be done. It would also be interesting to include other classification algorithms and compare their performances.

7. References:

[1] Nick Littlestone (1988). "Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm", Machine Learning 285–318(2)

[2] Lewis D.D. (1998) Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg

[3] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[4] Stone, M (1977). "An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion". Journal of the Royal Statistical Society: Series B (Methodological). **39** (1): 44–47. JSTOR 2984877.}