A NEW SUPERVISED LEARNING ALGORITHM USING NAÏVE BAYESIAN CLASSIFIER

Dewan Md. Farid, Jerome Darmont, and Nouria Harbi
Labratoire ERIC, Universite Lumiere Lyon 2
5 av. Pierre Mendes, France – 69676 BRON Cedex-France

Chowdhury Mofizur Rahman
Dept. of Computer Science & Engineering, United International University
UIU Bhaban, H#80, R#8/A, Dhanmondi, Dhaka-1209, Bangladesh

ABSTRACT

A new supervised learning algorithm using naïve Bayesian classifier is presented in this paper, which calculates the prior and conditional probabilities from a given training data and classifies the training examples using these probabilities. If any training example is misclassified then the algorithm calculates the information gain of attributes of the training data and chooses one attribute from training data with maximum information gain value. After the algorithm splits the training data into sub-datasets depending on the attribute values of the selected attribute, and again calculates the prior and conditional probabilities for each sub-dataset and classifies the examples of the each sub-dataset using their respective probabilities. The process will continue until all the training examples are correctly classified. Finally, the algorithm preserves the probabilities of each dataset for the future classification of unknown examples, whose attributes value are known but class value is unknown. The proposed algorithm addresses the problem of classifying the large dataset and it has been successfully tested on a number of benchmark problems, which achieved high classification rates using limited computational resources.

KEYWORDS

Conditional probabilities, classification rates, information gain, naïve Bayesian classifier.

1. INTRODUCTION

The naïve Bayesian classifier (NB) is one of the most popular data mining algorithms for classifying the large dataset. It has been successfully applied to the different problem domains of classification task such as intrusion detection, image and pattern recognition, weather forecasting, medical diagnosis, loan approval and bioinformatics etc [Hastie 2001], [Duda 2000], [Chai 2002]. The classification task is to map the set of attributes of sample data onto a set of class labels, and naïve Bayesian classifier particularly suitable as proven universal approximates. The NB classifier is a probabilistic approach for performing supervised learning that provides an optimal way to predict the class of an unknown example [Kononenko 1990], [Langely 1992]. In NB class conditional probabilities for each attribute values are calculated from the given training data, and then these probabilities are used to classify the known or unknown examples.

In this paper, we proposes a new learning algorithm which calculates the prior and conditional probabilities from a given training data and classifies the training examples using these probabilities, if any training example is misclassified then the algorithm calculates the information gain for attributes of training data and chooses one attribute of training data with maximum information gain value. After the algorithm splits the dataset into sub-datasets depending on the attribute values of chosen attribute, and again calculates the prior and conditional probabilities for each sub-dataset and classifies the examples of the each sub-dataset using their respective probabilities. The algorithm will continue this process until all the training examples are correctly classified. Finally, the algorithm preserves the prior and conditional probabilities of each dataset for the future classification of unknown examples. The attributes value of unknown examples is known, but class value is unknown. We tested our proposed algorithm on a number of benchmark datasets and the experimental result shows that our proposed algorithm achieved high classification rates compared to the existing naive Bayesian classifier on different problem domains.

The remainder of this paper is organized as follows. Section 2 describes the naïve Bayesian classifier. Section 3 provides our proposed algorithm. Section 4 provides the experiment analysis with benchmark datasets. Finally, section 5 makes some concluding remarks along with suggestions for further works.

2. NAÏVE BAYESIAN CLASSIFIER

Naïve Bayesian (NB) classifier is a simple probabilistic classifier based on probability model, which can be trained very efficiently in a supervised learning [Margaret 2005]. The NB classifier is given as input a set of training examples each of which is described by attributes A_1 through A_k and an associated class, C. The objective is to classify an unseen example whose class value is unknown but values for attributes A_1 through A_k are known and they are $a_1, a_2, \ldots, a_k$ respectively. The optimal prediction of the unseen example is the class value c such that $P(C=c_i|A_1=a_1,\ldots A_k=a_k)$ is maximum. By Bayes rule this probability equals to:

$$\operatorname{argmax}_{c_i \in C} \frac{P(A_1=a_1,..A_k=a_k \mid C=c_i)}{P(A_1=a_1,...A_k=a_k)} P(C=c_i) \qquad (1)$$

Where, $P(C=c_i)$ is the prior probability of class c_i, $P(A_1=a_1,\ldots A_k=a_k)$ is the probability of occurrence of the description of a particular example, and $P(A_1=a_1,\ldots A_k=a_k|C=c_i)$ is the class conditional probability of the description of a particular example c_i of class C. The prior probability of a class can be estimated from training data. The probability of occurrence of the description of particular examples is irrelevant for decision making since it is the same for each class value c. Learning is therefore reduced to the problem of estimating the class conditional probability of all possible description of examples from training data. The class conditional probability can be written in expanded from as follows:

$$\begin{aligned} &P(A_1=a_1,\ldots A_k=a_k|C=c_i) \\ &\quad = P(A_1=a_1|\ A_2=a_2 \wedge \ldots A_k=a_k \wedge C=c_i) \\ &\qquad * P(A_2=a_2|\ A_3=a_3 \wedge \ldots A_k=a_k \wedge C=c_i) \\ &\qquad\quad * P(A_3=a_3|\ A_4=a_4 \wedge \ldots A_k=a_k \wedge C=c_i) \\ &\qquad\qquad * P(A_4=a_4 \wedge \ldots A_k=a_k \wedge C=c_i) \end{aligned} \qquad (2)$$

In NB, it is assumed that outcome of attribute A_i is independent of the outcome of all other attributes A_j, given c. Thus class conditional probabilities become: $P(A_1=a_1,\ldots A_k=a_k|C=c_i) = \prod_{i=1}^{k} P(A_i = a_i \mid C = c_i)$ If the above value is inserted in equation "1" it becomes:

$$\equiv \arg\max_{c_i \in C} P(C=c) \prod_{i=1}^{k} P(A_i = a_i \mid C = c_i) \qquad (3)$$

In Naïve Bayesian classifier, the probability values of equation "3" are estimated from the given training data. These estimated values are then used to classify unknown examples.

3. PROPOSED LEARNING ALGORITHM

Given a training data, the proposed algorithm estimates the prior probabilities $P(C_j)$ for each class by counting how often each class occurs in the training data and conditional probabilities $P(A_{ij}|C_j)$ for each attribute values A_{ij} from the training data by counting how often each value occurs in the class in training data. After calculating prior and conditional probabilities the algorithm classifies the training examples using these prior and conditional probabilities. When classifying an example in the training data, the prior and conditional probabilities generated from the training data are used to make the prediction. This is done by combining the effects of the different attribute values from the training examples. Suppose the training example e_i has independent attribute values $\{A_{i1}, A_{i2},\dots,A_{ip}\}$, we know $P(A_{ik} \mid C_j)$, for each class C_j and attribute A_{ik}. We then estimate $P(e_i \mid C_j)$ by

$$P(e_i \mid C_j) = P(C_j) \prod_{k=1\to p} P(A_{ij} \mid C_j) \qquad (4)$$

To classify the example, we can estimate the likelihood that e_i is in each class. The probability that e_i is in a class is the product of the conditional probabilities for each attribute value with prior probability for that class. The posterior probability $P(C_j \mid e_i)$ is then found for each class. Then the example classifies with the highest posterior probability for that training example. If any training example is misclassified, then the algorithm calculates the information gain for attributes A_i in the training data. After calculating the information gain values for attributes in training data the algorithm chooses one of the one best attribute from the training data with the highest information gain value and splits the training data into sub-datasets depending on the attribute values of selected attribute. The algorithm again estimates the prior and conditional probabilities for each sub-dataset and classifies the examples of sub-datasets using their respective probabilities. If any example of a sub-dataset is misclassified then the algorithm calculates the information gain of attributes for that sub-dataset and chooses one attribute with highest information gain value and splits the sub-dataset into sub-sub-datasets. The algorithm will continue this process until all the examples of training data are correctly classified. When the algorithm correctly classifies all the training examples, then the algorithm terminates and the prior and conditional probabilities for each sub/sub-datasets are preserved for future classification of unseen examples. The main procedure of proposed algorithm is described as follows.

Algorithm
Input: Training Data, *D*
Output: Classification Model
Procedure:

1. Calculate the prior probabilities $P(C_j)$ for each class C_j from D by counting how often C_j occurs in D.
2. Calculate the conditional probabilities $P(A_{ij}|C_j)$ for each attribute values A_{ij} from D by counting how often each A_{ij} value occurs in the class in D.
3. Classify the examples in D with the highest posterior probability, $P(e_i \mid C_j) = P(C_j) \prod P(A_{ij} \mid C_j)$.
4. If any example in D is misclassified then calculate the information gain for attributes A_i in D.

   $$\text{Information Gain } (D, C) = H(D) - \sum_{i=1}^{C} P(D_i)H(D_i)$$

5. Select one best attribute A_i in D with maximum information gain value.
6. Split the dataset D into sub-datasets D_i based on attribute values A_{ij} of selected attribute A_i.
7. Continue step 1 to 5 until all the examples in D are correctly classified.
8. Preserved all the prior and conditional probabilities for each sub/sub-datasets D_i for future classification of unseen examples.

4. EXPERIMENT WITH BANCHMARK DATASET

First we tested our proposed algorithm on Tic-Tac-Toe benchmark dataset [Tic-Tac-Toe 1991], which encodes the complete set of possible board configurations at the end of Tic-Tac-Toe games, where "x" is assumed to play first. The target concept is "win of x" (i.e., true when "x" has one of 8 possible ways to create a "three-in-a-row"). In Tic-Tac-Toe dataset, there are total 958 instances/examples (626 positive examples and 332 negative examples), number of classes: 2 (positive and negative), and the number of attributes: 9 (each attribute corresponding to one tic-tac-toe square and has 3 attribute values x, o, and b), which follows:

1. A1= top-left-square: {x,o,b}
2. A2= top-middle-square: {x,o,b}
3. A3= top-right-square: {x,o,b}
4. A4= middle-left-square: {x,o,b}
5. A5= middle-middle-square: {x,o,b}

6. A6= middle-right-square: {x,o,b}
7. A7= bottom-left-square: {x,o,b}
8. A8= bottom-middle-square: {x,o,b}
9. A9= bottom-right-square: {x,o,b}

The prior and conditional probabilities are calculated for each class and each attribute values using the Tic-Tac-Toe dataset, which are shown in table 1 and table 2.

Table 1 Prior probabilities of Tic-Tac-Toe dataset

Prior probabilities	Value
P(Class=Positive)	0.65344
P(Class=Negative)	0.34655

Table 2 Conditional probabilities for each attribute values in Tic-Tac-Toe dataset

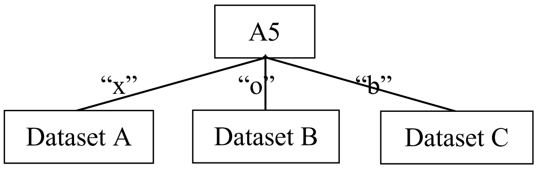
Conditional Probabilities	Value	Conditional Probabilities	Value
P(A1=x | Class=Positive)	0.47124	P(A2=x | Class=Positive)	0.35942
P(A1=x | Class=Negative)	0.37048	P(A2=x | Class=Negative)	0.46804
P(A1=o | Class=Positive)	0.30191	P(A2=o | Class=Positive)	0.36581
P(A1=o | Class=Negative)	0.43975	P(A2=o | Class=Negative)	0.30421
P(A1=b | Class=Positive)	0.22683	P(A2=b | Class=Positive)	0.27476
P(A1=b | Class=Negative)	0.18975	P(A2=b | Class= Negative)	0.23493
P(A3=x | Class=Positive)	0.47124	P(A4=x | Class=Positive)	0.35942
P(A3=x | Class= Negative)	0.37048	P(A4=x | Class= Negative)	0.46084
P(A3=o | Class=Positive)	0.30191	P(A4=o | Class=Positive)	0.36581
P(A3=o | Class= Negative)	0.43975	P(A4=o | Class= Negative)	0.30421
P(A3=b | Class=Positive)	0.22683	P(A4=b | Class=Positive)	0.27476
P(A3=b | Class= Negative)	0.18975	P(A4=b | Class= Negative)	0.23493
P(A5=x | Class=Positive)	0.58466	P(A6=x | Class=Positive)	0.35942
P(A5=x | Class= Negative)	0.27710	P(A6=x | Class= Negative)	0.46084
P(A5=o | Class=Positive)	0.23642	P(A6=o | Class=Positive)	0.36581
P(A5=o | Class= Negative)	0.57831	P(A6=o | Class= Negative)	0.30421
P(A5=b | Class=Positive)	0.17891	P(A6=b | Class=Positive)	0.27476
P(A5=b | Class= Negative)	0.14457	P(A6=b | Class= Negative)	0.23493
P(A7=x | Class=Positive)	0.47124	P(A8=x | Class=Positive)	0.35942
P(A7=x | Class= Negative)	0.37048	P(A8=x | Class= Negative)	0.46084
P(A7=o | Class=Positive)	0.30191	P(A8=o | Class=Positive)	0.36581
P(A7=o | Class= Negative)	0.43975	P(A8=o | Class= Negative)	0.30421
P(A7=b | Class=Positive)	0.22683	P(A8=b | Class=Positive)	0.27476
P(A7=b | Class= Negative)	0.18975	P(A8=b | Class= Negative)	0.23493
P(A7=x | Class=Positive)	0.47124	P(A8=x | Class=Positive)	0.35942
P(A9=x | Class=Positive)	0.47124	P(A9=o | Class= Negative)	0.43975
P(A9=x | Class= Negative)	0.37048	P(A9=b | Class=Positive)	0.22683
P(A9=o | Class=Positive)	0.30191	P(A9=b | Class= Negative)	0.18975

Then classify all 958 examples of Tic-Tac-Toe dataset using the prior and conditional probabilities of table 1 and table 2. Among 958 examples, total 943 examples are correctly classified, but 15 examples are misclassified. So, calculate the information gains of 9 attributes in Tic-Tac-Toe dataset [info_Total = 0.93, info_A1 = 0.91, info_A2 = 0.92, info_A3 = 0.91, info_A4 = 0.92, info_A5 = 0.84, info_A6 = 0.96, info_A7 = 0.91, info_A8 = 0.92, and info_A9 = 0.91]. Therefore, the information gains (info gain = info_Total – info_A) of 9 attributes are A1= 0.02, A2= 0.01, A3= 0.02, A4= 0.01, A5= 0.09, A6= -0.03, A7= 0.02, A8= 0.01, and A9= 0.02. The information gain value of attribute A5 is greater than other attributes. So, the Tic-Tac-Toe dataset will be divided into 3 sub-datasets A, B, and C as attribute A5 has 3 attribute values x, o, and b. Table 3 provides the details of Dataset A, B, and C.

Table 3 Details of Dataset A, B, and C

Dataset	Attribute Value	Total Examples	Positive Examples	Negative Examples
Dataset A	A5= “x”	458	366	92
Dataset B	A5= “o”	340	148	192
Dataset C	A5= “b”	160	112	48

Figure 1 Tree using attribute A5 of Tic-Tac-Toe dataset



Then the prior and conditional probabilities are calculated for each sub-dataset Dataset A, B, and C, that are shown in table 4 and table 5.

Table 4 Prior probabilities of Dataset A, B, and C

Prior probabilities	Dataset A	Dataset B	Dataset C
P(C=Positive)	0.79912	0.43529	0.7
P(C=Negative)	0.20087	0.56470	0.3

Table 5 Conditional probabilities of Dataset A, B, and C

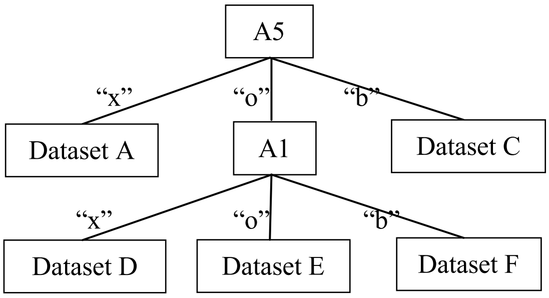
Conditional Probabilities	Dataset A	Dataset B	Dataset C	Conditional Probabilities	Dataset A	Dataset B	Dataset C
P(A1=x | C=Positive)	0.39344	0.58783	0.57142	P(A2=x | C=Positive)	0.34426	0.41216	0.33928
P(A1=x | C=Negative)	0.30434	0.42187	0.29166	P(A2=x | C=Negative)	0.36956	0.48958	0.52083
P(A1=o | C=Positive)	0.34972	0.22297	0.25	P(A2=o | C=Positive)	0.38797	0.27702	0.41071
P(A1=o | C=Negative)	0.54347	0.36458	0.54166	P(A2=o | C=Negative)	0.35869	0.28645	0.27083
P(A1=b | C=Positive)	0.25683	0.18918	0.17857	P(A2=b | C=Positive)	0.26775	0.31081	0.25
P(A1=b | C=Negative)	0.15217	0.21354	0.16666	P(A2=b | C= Negative)	0.27173	0.22395	0.20833
P(A3=x | C=Positive)	0.39344	0.58783	0.57142	P(A4=x | C=Positive)	0.34426	0.41216	0.33928
P(A3=x | C= Negative)	0.30434	0.42187	0.29166	P(A4=x | C= Negative)	0.36956	0.48958	0.52083
P(A3=o | C=Positive)	0.34972	0.22297	0.25	P(A4=o | C=Positive)	0.38797	0.27702	0.41071
P(A3=o | C= Negative)	0.54347	0.36458	0.54166	P(A4=o | C= Negative)	0.35869	0.28645	0.27083
P(A3=b | C=Positive)	0.25683	0.18918	0.17857	P(A4=b | C=Positive)	0.26775	0.31081	0.25
P(A3=b | C= Negative)	0.15217	0.21354	0.16666	P(A4=b | C= Negative)	0.27173	0.22395	0.20833
P(A6=x | C=Positive)	0.34426	0.41216	0.33928	P(A7=x | C=Positive)	0.39344	0.58783	0.57142
P(A6=x | C= Negative)	0.36956	0.48958	0.52083	P(A7=x | C= Negative)	0.30434	0.42187	0.29166
P(A6=o | C=Positive)	0.38797	0.27702	0.41071	P(A7=o | C=Positive)	0.34972	0.22297	0.25
P(A6=o | C= Negative)	0.35869	0.28645	0.27083	P(A7=o | C= Negative)	0.54347	0.36458	0.54166
P(A6=b | C=Positive)	0.26775	0.31081	0.25	P(A7=b | C=Positive)	0.25683	0.18918	0.17857
P(A6=b | C= Negative)	0.27173	0.22395	0.20833	P(A7=b | C= Negative)	0.15217	0.21354	0.16666
P(A8=x | C=Positive)	0.34426	0.41216	0.33928	P(A9=x | C=Positive)	0.39344	0.58783	0.57142
P(A8=x | C= Negative)	0.36956	0.48958	0.52083	P(A9=x | C= Negative)	0.30434	0.42187	0.29166
P(A8=o | C=Positive)	0.38797	0.27702	0.41071	P(A9=o | C=Positive)	0.34972	0.22297	0.25
P(A8=o | C= Negative)	0.35869	0.28645	0.27083	P(A9=o | C= Negative)	0.54347	0.36458	0.54166
P(A8=b | C=Positive)	0.26775	0.31081	0.25	P(A9=b | C=Positive)	0.25683	0.18918	0.17857
P(A8=b | C= Negative)	0.27173	0.22395	0.20833	P(A9=b | C= Negative)	0.15217	0.21354	0.16666

Now classify all examples of Dataset A, B, and C with their respective probabilities. All examples of Dataset A and C are correctly classified. But for Dataset B 254 examples are correctly classified and 86 examples are misclassified. So, again calculate the information gain of attributes in Dataset B [info_ B_Total = 0.987, info_ B_A1 = 0.954, info_B_A2 = 0.976, info_B_A3 = 0.954, info_B_A4 = 0.976, info_B_A6 = 0.976, info_B_A7 = 0.954, info_B_A8 = 0.976, and info_B_A9 = 0.954]. So the information gains of 8 attributes in Dataset B are B_A1= 0.033, B_A2= 0.011, B_A3= 0.033, B_A4= 0.011, B_A6= 0.011, B_A7= 0.033, B_A8= 0.011, and B_A9= 0.033. The information gains of attributes A1, A3, A7, and A9 in Dataset B are equal, also maximum than other attributes. So, we consider A1 and the Dataset B will be divided into 3 sub-sub-datasets D, E, and F as attribute A1 has 3 attribute values x, o, and b. Table 6 provides the details of Dataset D, E, and F.

Table 6 Details of Dataset D, E, and F

Dataset	Attribute Value	Total Examples	Positive Examples	Negative Examples
Dataset D	A1= “x”	168	87	81
Dataset E	A1= “o”	103	33	70
Dataset F	A1= “b”	69	28	41

Figure 2 Tree after dividing Dataset B



Again the prior and conditional probabilities are calculated using the Dataset D, E, and F that are shown in table 7 and table 8.

Table 7 Prior probabilities of Dataset D, E, and F

Prior Probabilities	**Dataset D**	**Dataset E**	**Dataset F**
P(Class=Positive)	0.51785	0.32038	0.40579
P(Class=Negative)	0.48214	0.67961	0.59420

Table 8 Conditional probabilities of Dataset D, E, and F

Conditional Probabilities	**Dataset A**	**Dataset B**	**Dataset C**	**Conditional Probabilities**	**Dataset A**	**Dataset B**	**Dataset C**
P(A2=x | C=Positive)	0.52873	0.30303	0.17857	P(A3=x | C=Positive)	0.57471	0.63636	0.57142
P(A2=x | C=Negative)	0.37037	0.61428	0.51219	P(A3=x | C= Negative)	0.30864	0.55714	0.41463
P(A2=o | C=Positive)	0.22988	0.27272	0.42857	P(A3=o | C=Positive)	0.24137	0.15151	0.25
P(A2=o | C=Negative)	0.38271	0.15714	0.31707	P(A3=o | C= Negative)	0.48148	0.2	0.41463
P(A2=b | C=Positive)	0.24137	0.42424	0.39285	P(A3=b | C=Positive)	0.18390	0.21212	0.17857
P(A2=b | C= Negative)	0.24691	0.22857	0.17073	P(A3=b | C= Negative)	0.20987	0.24285	0.17073
P(A4=x | C=Positive)	0.52873	0.30303	0.17857	P(A6=x | C=Positive)	0.26436	0.63636	0.60714
P(A4=x | C= Negative)	0.37037	0.61428	0.51219	P(A6=x | C= Negative)	0.41975	0.6	0.43902
P(A4=o | C=Positive)	0.22988	0.27272	0.42857	P(A6=o | C=Positive)	0.35632	0.15151	0.17857
P(A4=o | C= Negative)	0.38271	0.15714	0.31707	P(A6=o | C= Negative)	0.37037	0.14285	0.36585
P(A4=b | C=Positive)	0.24137	0.42424	0.39285	P(A6=b | C=Positive)	0.37931	0.21212	0.21428
P(A4=b | C= Negative)	0.24691	0.22857	0.17073	P(A6=b | C= Negative)	0.20987	0.25714	0.19512
P(A7=x | C=Positive)	0.57471	0.63636	0.57142	P(A8=x | C=Positive)	0.26436	0.63636	0.60714
P(A7=x | C= Negative)	0.30864	0.55714	0.41463	P(A8=x | C= Negative)	0.41975	0.6	0.43902
P(A7=o | C=Positive)	0.24137	0.15151	0.25	P(A8=o | C=Positive)	0.35632	0.15151	0.17857
P(A7=o | C= Negative)	0.48148	0.2	0.41463	P(A8=o | C= Negative)	0.37037	0.14285	0.36585
P(A7=b | C=Positive)	0.18390	0.21212	0.17857	P(A8=b | C=Positive)	0.37931	0.21212	0.21428
P(A7=b | C= Negative)	0.20987	0.24285	0.17073	P(A8=b | C= Negative)	0.20987	0.25714	0.19512
P(A9=x | C=Positive)	0.29885	1.0	1.0	P(A9=o | C= Negative)	0.18518	0.71428	0.12195
P(A9=x | C= Negative)	0.51851	0.21428	0.58536	P(A9=b | C=Positive)	0.32183	0.0	0.0
P(A9=o | C=Positive)	0.37931	0.0	0.0	P(A9=b | C= Negative)	0.29629	0.07142	0.29268

Now all examples of the Dataset D, E, and F are correctly classified using their respective probabilities and finally the probabilities for each dataset A, C, D, E, and F are saved for future classification of known or unknown examples.

4.1 Experimental Analysis

To evaluate the performance of our proposed algorithm with naïve Bayesian classifier on different problem domains, we performed experiments on 7 datasets as obtained from UCI repository that provided in table 9. The experimental results in table 9 illustrate that our proposed algorithm achieved better classification rates than naïve Bayesian classifier.

Table 9 Classification rates (%) for NB classifier and proposed algorithm

Dataset	No. of Class	No. of Attributes	No. of Cases	NB classifier	Proposed algorithm
Tic-Tac-Toe	2	9	958	98.43	100
Soybean	19	35	683	97.65	99.35
Iris	3	4	151	92.43	96.55
Monk 3	2	6	554	93.11	97.20
Zoo	7	16	101	98.69	99.47
Diabetes	2	8	768	96.70	98.27
Vehicle	2	18	946	94.68	99.19

5. CONCLUSION

In this paper we have concentrated on the development of the classification rates for naïve Bayesian classifier by splitting the training data into sub-datasets based on the best attributes of training data, which improves the prior and conditional probabilities value. The naïve Bayesian classifier has several advantages such as it is easy to use and only one scan of training data is required. The naïve Bayesian classifier can easily handle the missing values by simply omitting the probability when calculating the likelihoods of membership in each class. The future research issue will be to build a hybrid supervised learning algorithm by merging with other data mining algorithms and also apply this algorithm in real world problem domains.

ACKNOWLEDGEMENT

Support for this research received from ERIC Laboratory, University Lumière Lyon 2 – France, and Department of Computer Science and Engineering, United International University, Bangladesh.

REFERENCES

Margaret H. Dunham, 2005. *Data Mining: Introductory and Advanced Topics*. Pearson Education Pte. Ltd., Singapore.

T. Hastie et al, 2001, The Elements of Statistical Learning: Data Mining, Inference and Prediction, *Heidelberg, Germany: Springer-Verlag*.

R. O. Duda et al, 2000, Pattern Classification, *2nd ed. Chichester, U.K.: Wiley-Interscience*.

K. M. A. Chai et al, 2002, Bayesian online classifiers for text classification and filtering, *Proceedings of SIGIR 2002,* Tampere, Finland, pp. 97-104.

R. Jin and G. Agrawal, 2003, Efficient decision tree construction on streaming data, *Proceedings of ACM SIGKDD*, pp 571-576.

L. Breiman et al, 1993, Classification and Regression Trees, *Boca Raton, FL: Chapman & Hall*.

Kononenko I, 1990, Comparison of inductive and naïve Bayesian learning approaches to automatic knowledge acquisition, *Wieling, B. (Ed), Current trend in knowledge acquisition*, Amsterdam, IOS press.

Langely, P. et al, 1992, An analysis of Bayesian classifier, *Proceedings of the 10th national Conference on Artificial Intelligence* (San Matro, CA: AAAI press), pp. 223-228.

The Tic-Tac-Toe Dataset. 1991 http://archive.ics.uci.edu/ml/datasets/Tic-Tac-oe+Endgame

The Archive UCI Machine Learning Datasets. http://archive.ics.uci.edu/ml/datasets/