

Optimized Naïve Bayesian Algorithm for Efficient Performance

Obuandike Georgina N.
Department of Mathematical Sciences and IT, Federal University Dutsinma, Katsina state, Nigeria.
P.M.B 5001, Dutsinma, Katsina State

Audu Isah
Department of Mathematics and Statistics, Federal University of Technology, Minna, Niger State

John Alhasan
Department of Computer Science, Federal University of Technology, Minna, Niger State, Nigeria

Abstract
Naïve Bayesian algorithm is a data mining algorithm that depicts relationship between data objects using probabilistic method. Classification using Bayesian algorithm is usually done by finding the class that has the highest probability value. Data mining is a popular research area that consists of algorithm development and pattern extraction from database using different algorithms. Classification is one of the major tasks of data mining which aimed at building a model (classifier) that can be used to predict unknown class labels. There are so many algorithms for classification such as decision tree classifier, neural network, rule induction and naïve Bayesian. This paper is focused on naïve Bayesian algorithm which is a classical algorithm for classifying categorical data. It easily converged at local optima. Particle Swarm Optimization (PSO) algorithm has gained recognition in many fields of human endeavours and has been applied to enhance efficiency and accuracy in different problem domain. This paper proposed an optimized naïve Bayesian classifier using particle swarm optimization to overcome the problem of premature convergence and to improve the efficiency of the naïve Bayesian algorithm. The classification result from the optimized naïve Bayesian when compared with the traditional algorithm showed a better performance
Keywords: Data Mining, Classification, Particle Swarm Optimization, Naïve Bayesian

1.Introduction

Classification is basic task in data mining and machine learning which aimed at building a classification model using dataset attributes (Zang, 2004; Rennie et al, 2003). A classification models is a function that assigns class labels to a dataset. It predicts class labels $A \in (A_1 \dots A_m)$ for a set of test data $T = (T_1 \dots T_n)$. Naïve Bayes is a popular classification algorithm that is quite effective and efficient for classification of categorical data. Its performance in classification task is quite impressive but its attribute independent assumption technique on which it operates is usually a challenge to its methodology (Zang, 2004). Thus, it is usually applied as a baseline in classification task because of its speed and simplicity and usually placed last among other classification algorithms, yet it is most frequently used for classification of text data because of its simplicity and easy to use nature. The other preferred or higher algorithms are usually slower and more complex (Rennie et al, 2003).

Several methods have been applied in order to enhance the performance of Naïve Bayes algorithm in order to handle the issue of attribute independent assumption (Jiang et al, 2007). In this work optimized Naïve Bayes was proposed using Particle Swarm Optimization (PSO). The fitness function was modelled based on the structure of Naïve Bayes algorithm. The optimal value was determined by the PSO technique. Thus, Naïve Bayes accuracy was used as the fitness function. The rest of the section is discussion on Naïve Bayes algorithm, PSO, methodology for the optimization of Naïve Bayes and result discussion.

2. Naïve Bayes

Naïve Bayes is a statistical classification algorithm that uses Bayes theorem which was named after the author Thomas Bayes (Jiawei et al, 2012). It is a popular classifier among other classifiers such as decision tree and Support Vector Machine. Despite its simplicity, it competes favourably with other classifiers (Taheri et al, 2015). Suppose T is a dataset with associated $n -$ dimensional attributes, each record in the dataset can be represented as $R = (r_1, r_2, \dots r_n)$ consist of a set of records with n-dimensional attributes $F_1, F_2, \dots F_3$. If there p classes $K_1, K_2, \dots K_p$ given a set of records R. The classifier can predict that the record r_i belongs to class k_i when:

$$P(K_i \backslash R) > P(K_j \backslash R) \text{ for } 1 \leq j \leq p,\ j \neq i \qquad (1)$$

When $P(K_i \backslash R)$ is maximized, the maximum posterior probability for a set of records is thereby given as:

$$P(K_i \backslash R) = \frac{P(R \backslash K_i) P(K_i)}{P(R)} \qquad (2)$$

If $P(R \backslash K_i)$ is maximized, the evaluation of $P(R \backslash K_i)$ for a dataset with high dimension is difficult, thus the naïve assumption of attribute independent is made thus,

$$P(R\backslash K_i) = \prod_{q=1}^{n} P(R_q\backslash K_i) = P(r_1\backslash K_i) \times P(r_2\backslash K_1) \times \ldots \times P(r_n\backslash K_i) \quad (3)$$

The classifier of Naïve Bayes can be represented as

$$N_b = \arg\max P(K_i) \prod_{q=1}^{n} P(R_q\backslash K_i), \quad k_i \in K \quad (4)$$

This describes the most probable class value. The Bayesian strategy is to assign the most probable class to the test record.

3. Related Works

The methodology of the attribute independence assumption made by the Naïve Bayes classifier sometimes affects its classification performance when it is violated. In order to solve this problem attribute independent assumption of the Naïve Bayes algorithm and at the same time maintain its simplicity and efficiency, many researchers have proposed different methods that can improve the performance of the Naive Bayes classifier by handling the attribute independence assumption. The works include the work that was named semi Naïve Bayes which used some selected subsets of attributes and performance evaluation showed great improvement in the Naïve Bayes (Kohavi, 1996; Pazzani, 1996). The other work utilized tree structure to find out relationship between tuples (Friedman, 1997). Taheri et al (2011) in their work used conditional probabilities to find the relationships between attributes. Other works done were mostly on weighted attributes where weights were assigned to attributes according to order of importance since attributes do not have equal impact on dataset (Ozsen and Gunecs, 2009; Hall, 2007; Zhou and Huang, 2006). So many works were also done in order to evaluate important of attributes in a dataset (Ozsen and Gunecs, 2009; Hall, 2007; Zhou and Huang, 2006).

Jiang and Zhang (2007) in their work improved Naïve Bayes termed weighted averaged dependence estimators used the idea introduced by Webb et al (2005). Hall (2007) in his work Decision tree attribute filter is of the opinion that weight given to a class attribute should be the inverse of the degree of its dependency on other attributes. Wu and Cai (2011) recently applied differential evolution algorithms to assign weights to attributes which was used to develop the weighted naïve bayes classifier. Zang and Sheng (2005) investigated how to use the assigned weights of Naïve bayes to achieve accurate ranking from dataset. In this work, an optimized naïve bayes is proposed which applied PSO technique to improve the performance to the Naïve Bayes algorithm.

4. Particle Swarm Optimization

A conventional PSO looks at a swarm S that comprises of particles in a multidimensional space of n-dimension and every ith particle in the group has a position denoted as $X_i = (x_{i1}, x_{i2}, \ldots\ x_{in})$ and a velocity denoted as $V_i = (v_{i1}, v_{i2}, \ldots\ v_{in})$. The position of each particle is usually changed using equation 5.

$$X_{i+1} = x_i + v_i \quad (5)$$

In real life, birds usually change their positions for a better position based on their personal experiences and the experiences of other birds in the group. Likewise, in PSO application calculation of each particle also changes its velocity to show the nature of the new position. The individual position is represented as $Pb_i = (Pb_{i1}, Pb_{i2}, \ldots\ Pb_{in})$ and general position is represented as $Gb_i = (gb_{i1}, gb_{12}, \ldots\ gb_{in})$.
The change in velocity is calculated using equation 6.

$$V_{i+1} = (\omega V_i + C_1 r_1 (pb_i - X_i) + C_2 r_2 (gb_i - X_i) \quad (6)$$

Where V_i is the current velocity of the particles i while V_{i+1} is the new velocity that need to be achieved to be able to move from the current position to the new position. The range of velocities is bounded between V_{max} and V_{min}; where as V_{max} is the maximum velocity and V_{min} is the minimum velocity. The parameters c_1 and c_2 are the acceleration coefficients for social and cognitive components. The usual choice is to set C1 = C2 within the range [0,4]. r_1 and r_2 are two random numbers ranging from 0 to1 that determines the influence of pBest and gBest on the velocity update formula, and ω is the inertia of the particle which controls the momentum of the particle. Velocity added to the current position provides the new position of the particle as is given in Equation 5.

5. Data Description

Table 1: Description of the features of the dataset used

Features	Description
Education	• No Education: Implies not having any educational qualification • Low Education: includes Primary and Secondary graduates • Average Education: includes ND, NCE • High Education: includes Degrees, PGD, MSC, MA,PhD
Occupation	• Unemployed: implies no work at all • Self Employed: includes farmers, Apprentice, traders, artisans • Employed: includes private employed and government employed
Age	• Early Adulthood: ages 18 – 34 • Middle Adulthood: ages 35 – 50 • Late Adulthood: ages 51 – 150
Crime	• Low Crime: Breach of Trust, Conspiracy, Assult • Average Crime: Rape, Kidnapping, Drug • High Crime: Homicide, Armed Robbery, Theft
Sex	• Male • Female
Class	• Vulnerable • Non Vulnerable

6. Methodology

The PSO swarm size comprises of no of particles in the database (1733) and the particles are the number of features in the database (6). The weights and velocities of each particle was updated using equation 5 and 6 respectively.

In this work naïve bayes algorithm was optimized using PSO techniques respectively. The PSO algorithm used the classification accuracy of the naïve bayes as fitness functions. The velocities and weight of the features were updated using PSO algorithm formulas as shown in Equations 5 and 6 respectively.

The naïve bayes algorithms then run to give the accuracy again until the most optimal weight of the features are obtained. These finalized weights were then used to classify the test data respectively to obtain the final accuracy. The algorithms used the crime dataset to train and test models.

6.1 PSONaive Bayes Algorithm

In PSONaive Bayes 1433 instances were used to train the model while 300 intances were used to test the model. The algorithm for the PSONaive Bayes and flowchart are shown in Figure 1 and Figure 2

Algorithm

```
Parameters: No of particles N, r1, r2, w, c1, c2, Vmax, Vmin
Initialize the velocity and weight of the particle Vi(t) = 0 and Xi (t) = 0
Start
Accuracy = NaiveBayes (train data,1433)
While within bounded area
Begin
For I = 1 to n
fitnessfunction = Accuracy, set pBest and gBest
Update the velocity and position of the particles
Next i
End While
NaiveBayesian (test data,300)
Evaluate Classification
End
```

Figure 1. PSONaive Bayes Algorithm.

Where n is the total number of particles in the space. Together pBest and gBest were used to define the velocity of the particle which guides the particle towards a better solution

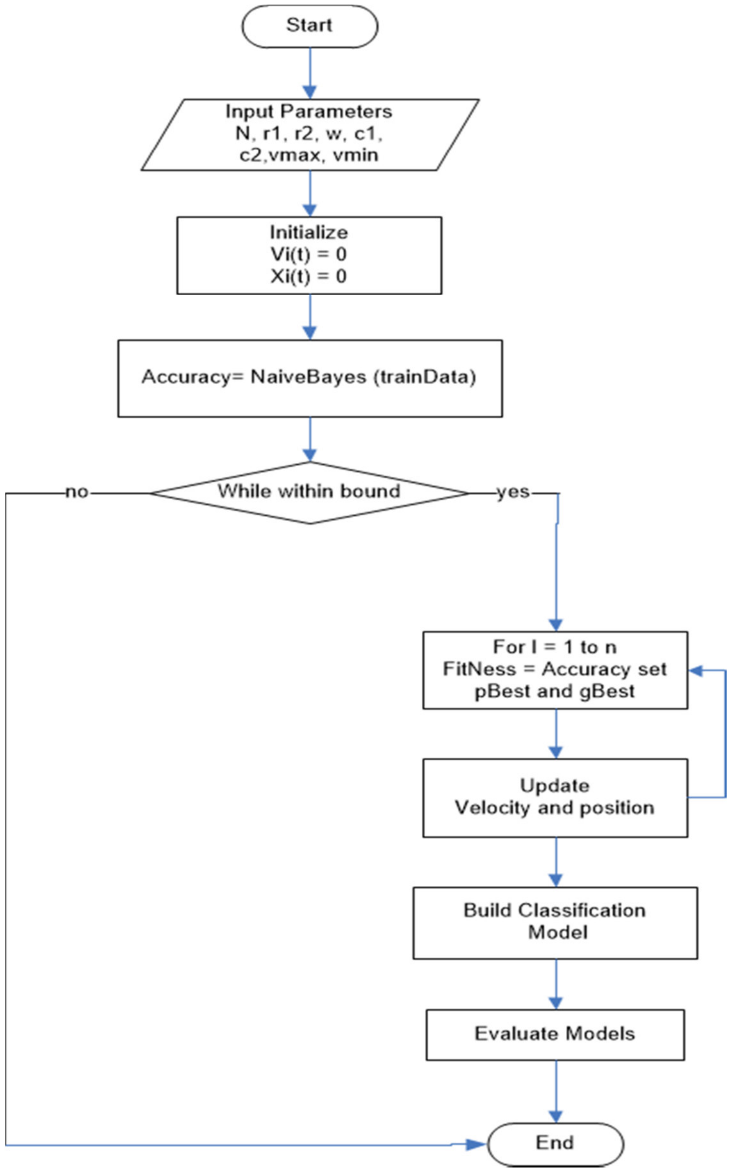

Figure 2: PSONaive Bayes Flowchart

6.2 Evaluation Metrics

The metrics used in the discussion of the result are as follows:
Accuracy: this shows the percentage of correctly classified instances in each classification model
Success Rate: is the statistics that shows correctly classified instances.
Failure Rate: is the report of instances incorrectly labeled as correct instances.
Time: time taken to build the models
ROC curve: is used to visualize classifiers performance. It is usually plotted using two metrics: TP Rate and FP Rate. The y axis is usually for TP Rate while x axis denotes the FP Rate. The ROC area is used to measure its performance. If the area is 1 it indicates perfect prediction, if it is 0.5 it implies random guess

7. Result Discussion

Table2: Tabulated Results of the performance of Naïve Bayes and the optimized Naïve Bayes

Parameters	Nb	psoNb
Sensitivity	99.5	100
Specificity	25	37.5
Accuracy	97	98
Time	1.844	1.14

The tabulated result in table 1 reveals that the optimized NaiveBayes algorithm gave higher accuracy of 98 and lesser time of 1.14 secs compared to ordinary NaiveBayes. The performance of Naïve Bayes and optimized

Naïve Bayes is represented graphical using ROC curve as shown in Figure 3.

7.1 ROC Curve of NaiveBayes and PSONaiveBayes Algorithms

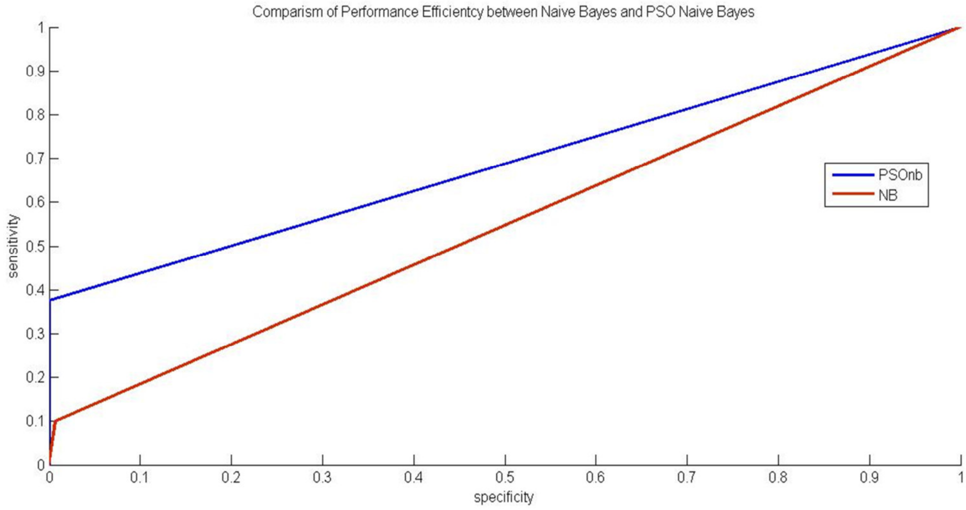

Figure 3: Comparison of Performance Efficiency of NaiveBayes and PSONaiveBayes Algorithms Using ROC Curve.

The ROC curve showed that PSONaiveBayes also Performs better than Ordinary NaiveBayes algorithm.

8.Conclusion

The increase in complexity and volumes of data recently been generated in various database repositories has called for an efficient data mining tools to improve mining efficiency and accuracy. Particle Swarm Optimization (PSO) is one of the Swarm Intelligence techniques whose algorithm has been proved to improve efficiency and accuracy in many area of human endeavour where it has been applied.

This work applied the PSO technique to improve the efficiency and accuracy of Naïve Bayesian classification technique. The evaluation result revealed that the optimizing the algorithm improved the classification accuracy from 97 percent to 98 percent. The ROC curve values which is used to measure performance of classification showed higher area under curve in optimized NaiveBayes than in ordinary NaiveBayes. The optimized algorithm when used for classification prove to have reduced the classification time from 1.93 secs to 1.14secs for.

The naiveBayes algorithm is a popular algorithm that handles categorical or non continuous data but has challenge in its methodology which attribute is attribute independent assumption and it usually being criticized for that methodology, this work tends to address it in order to improve its performance efficiency and accuracy.

References

Andreeva, P. (2006) “Data Modelling and Specific Rule Generation via Data Mining Techniques”, International Conference on Computer Systems and Technologies – CompSysTech, 2006.

Friedman N., Geiger D., Goldszmidt M. (1997) “Bayesian network classifiers”, Machine Learning 29, 131-163 (1997)

Hall M. (2007) “A Decision Tree-Based Attribute weighting Filter for Naive Bayes”, In Knowledge-Based Systems, Vol 20, 120-126

Han, j. and Kamber M. (2006) “Data Mining Concepts and Techniques”, Morgan Kaufmann Publishers

Jiang, D. Wang, Z. Cai, X. Yan (2007) “Survey of improving naive Bayes for classification”,In Proceedings of the 3rd International Conference on Advanced Data Mining and Applications, Springer, vol. 4632, 134-145

Kohavi R. (1996) “Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid”, In: Proc. 2nd ACM SIGKDD Int, Conf, Knowledge Discovery and Data Mining, 202-207

Ozsen S., Gunecs S. (2009), “Attribute weighting via genetic algorithms for attribute weighted artificial immune system (AWAIS) and its application to heart disease and liver disorders problems”, Expert Systems with Applications, 36, 386-392

Pazzani M. J. “(1996) Constructive induction of Cartesian product attributes”, ISIS: In- formation, Statistics and Induction in Science, 66-77

Rennie J. D. M., Shih L., Teevan J. and Karger D. R. (2003) “Tackling Poor Assumption of Naïve Bayes Text Classifier”, Proceedings of 20th International Conference on Machine Learning (ICML) Washington DC.

Sitar-Taut, V.A., et al., (2009) "Using machine learning algorithms in cardiovascular disease risk evaluation", Journal of Applied Computer Science & Mathematics,

Srinivas, K., Rani B. K., and Govrdhan A. (2010) "Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks". International Journal on Computer Science and Engineering (IJCSE), 2010. Vol. 02, No. 02: p. 250-255.

Taheri S., Mammadov M. (2011) "Tree Augmented Naive Bayes Based On Optimization" Proceeding of 42nd Annual Iranian Mathematics Conference, Vali-e-Asr University of Rafsanjan, Iran

Taheri S., Yearwood J., Mammadov M. and Seifollahi S. (2014) "Attribute Weighted Naïve Bayes Classifier Using a Local Optimization", Neural Computing and Application

Webb G. I., Boughton J., Wang Z. (2005) "Not so naive bayes: Aggregating one dependence Estimators", Machine Learning, Vol. 58, 5-24

Wu J., Cai Z. (2011) "Attribute Weighting via Differential Evolution Algorithm for Attribute Weighted Naive Bayes (WNB)", Journal of Computational Information Systems 7:5, 1672-1679

Zang H. (2004) "The Optimality of Naïve Bayes" American Association for Artificial Intelligence

Zhang H., Sheng S. (2005) "Learning weighted naive Bayes with accurate ranking", In Proceeding of the 4th IEEE International Conference on Data Mining, 567-570 (2005)

Zhou Y., Huang T. S. (2006) "Weighted Bayesian Network for Visual Tracking, Proceedings of the 18th International Conference on Pattern Recognition (ICPR'O6), 0-7695- 2521-0106