Boosted Gaussian Bayes Classifier and Its Application in Bank Credit Scoring

Pizzo ANAÏS [1,*], *Teyssere PASCAL* [2], *Long VU-HOANG* [3]

[1]Statistics and IT, Polytech Lille, Lille 1 University, Lille, France
[2]Statistics and IT, Polytech Lille, Lille 1 University, Lille, France
[3]VS Foods Joint Stock Company, Vietnam

*Corresponding Author: Pizzo ANAÏS or Teyssere PASCAL (email: anais.pizzo@polytech-lille.net, pascal.teyssere@polytech-lille.net)
(Received: 18-June-2018; accepted: 14-July-2018; published: 20-July-2018)
DOI: http://dx.doi.org/10.25073/jaec.201822.193

Abstract. *With the explosion of computer science in the last decade, data banks and networks management present a huge part of tomorrows problems. One of them is the development of the best classification method possible in order to exploit the data bases. In classification problems, a representative successful method of the probabilistic model is a Naïve Bayes classifier. However, the Naïve Bayes effectiveness still needs to be upgraded. Indeed, Naïve Bayes ignores misclassified instances instead of using it to become an adaptive algorithm. Different works have presented solutions on using Boosting to improve the Gaussian Naïve Bayes algorithm by combining Naïve Bayes classifier and Adaboost methods. But despite these works, the Boosted Gaussian Naïve Bayes algorithm is still neglected in the resolution of classification problems. One of the reasons could be the complexity of the implementation of the algorithm compared to a standard Gaussian Naïve Bayes. We present in this paper, one approach of a suitable solution with a pseudo-algorithm that uses Boosting and Gaussian Naïve Bayes principles having the lowest possible complexity.*

Keywords

Adaboost, Boosted Gaussian Naïve Bayes, Classification, Naïve Bayes

1. INTRODUCTION

In machine learning and statistics, classification is one of the most important tools to analyze and classify a large amount of data. Classification is the problem of identifying to which from several categories, a new observation belongs. Elkan C (1997) [1] and after him Ridgeway G, Madigan D, Richardson T, O'Kane J (1998) [2] presented the advantage of the Boosting methods and the interest of using it in classification problems. Many researchers have studied classification problems in order to improve the quality and efficiency of classification. An example would be assigning a given email to the "spam" or "non-spam" class or assigning a given Iris dataset into three main groups: Iris setosar, Iris versicolor, Iris virginica as detailed by characteristic of the Iris (sepal length, sepal width, petal length, petal width). Over the past few years, Naïve Bayes has had significant achievements in many practical applications, including medical diagnosis, systems performance management and text classification [3]-[7].

The family of Naïve Bayes classifier is commonly used as a probabilistic learning algorithm, by using the probability that a new observation belongs to a specific class. Naïve Bayes is based on Bayes' theorem. Naïve Bayes classifier is called Naïve because of an idealistic hypothesis that assumes the independence of the ran-

dom variables. Despite its Naïve hypothesis, the Naïve Bayes classifier is widely used due to its performance in real-world situations [8]-[12].

One technique to deal with continuous data is the Gaussian Naïve Bayes that assumes the continuous values associated with each class are distributed according to a Gaussian distribution parameterized by the corresponding means and standard derivations. Then it computes the posterior probability density function using normal distribution of classes. Because of its usability and flexibility, the Gaussian Naïve Bayes is applied in this article for dealing with continuous data.

Moreover, Naïve Bayes is applied in this article as it can combine observed data, previous knowledge and practical learning algorithm. However, the major limitation of Naïve Bayes classifier is that it ignores misclassified observations instead of adapting to tweak misclassified observations.

Adaptive Boosting or AdaBoost was an algorithm proposed by Freund and Schapire in 1995 [13]. Recently, it has been extensively used and studied in classification [14]-[16]. AdaBoost combines many "weak learners" into a weighted sum to get a "strong learner". Then, by increasing or decreasing the weighted sum of each "weak learner", Adaboost focuses on assigning instances misclassified by previous classifiers [13].

Boosted Gaussian Naïve Bayes is a new algorithm using both Gaussian Naïve Bayes classifier and AdaBoost's advantages. The basic idea is first to take an advantage of Gaussian Naïve Bayes that is easy to establish the discriminant function that plays a role "weak learners" in Adaboost between two categories. However, the discriminant function of Gaussian Naïve Bayes is nonlinear; hence, it is can be stronger than linear discriminant function created by Adaboost in most cases. Second, we use AdaBoost to adjust the weighted sum of observations of each weak learner and combine the weak learners to get the output. With this combination, the adaptability and efficiency of Gaussian Naïve Bayes can be increased by focusing on assigning observations that misclassified.

The rest of this article is organized as follows. Firstly, the Naïve Bayes classifier, the Gaussian Bayes classifier and the Ababoost algorithm are described. Secondly, the Boosted Gaussian Naïve Bayes is explained. Thirdly, numerical examples in the bank credit scoring field are analyzed. Finally, we conclude and future works are proposed.

2. PRELIMINARY

2.1. Naïve Bayes Classifier

The Naïve Bayes classifier is a classification method based on the Bayes Theorem Eq. (1):

$$P\left(A|B\right) = \frac{P\left(B|A\right)P(A)}{P(B)}. \tag{1}$$

In the case of classification, the Theorem can be interpreted with this approach:

- Let D be a training set of samples having the information about the classes. We consider m classes $w_1, w_2, \ldots, w_m$. And $X = X_1, X_2, \ldots, X_n$ with $x = x_1, x_2, \ldots, x_n$ be a specific sample of n value with n attribute.
- Given a sample x, the probability $P(w_i|x)$ is the posterior probability that x belongs to the class w_i. The classifier will affect x to the class w_i having the biggest $P(w_i|x)$.

According to the Bayes's theorem Eq. (2):

$$P\left(w_i|x\right) = \frac{P\left(x|w_i\right)P(w_i)}{P(x)} \tag{2}$$

- As $P(X)$ is the same for all classes and $P(w_i)$, the prior probability is the same for each data sample, we only need to compute $P(X|w_i)$.
- In order to reduce the computational cost, the most common approach is to estimate $P(X|w_i)$ instead of calculating it. By admitting that the classes belonging probabilities are independent, the formula can be calculated by Eq. (3):

$$P(X|w_i) \approx \prod_{k=1}^{n} P\left(x_k|w_i\right). \tag{3}$$

2.2. Gaussian Naïve Bayes classifier

The Gaussian Naïve Bayes classifier is a special case of Naïve Bayes in continuous case.

With k classes $w_1, w_2, \ldots, w_k$; with the prior probability q_i, $i = 1, k$; and $X = X_1, X_2, \ldots, X_n$ the n dimensional data sample.

In continuous case, $P(x|w_i)$ is calculated by Eq. (4):

$$P\left(w_i|x\right) = \frac{P(w_i)f\left(x|w_i\right)}{\sum_{i=1}^{n} P(w_i)f\left(x|w_i\right)} = \frac{q_i f_i(x)}{f(x)} \quad (4)$$

With:

- $P(w_i|x)$: the class a prior probability of class w_i,
- $f(x|w_i) = f_i(x)$: the probability density function of class w_i Eq. (5),
- $f\left(x\right) = q_i f_i\left(x\right) + q_j f_j\left(x\right)$, with $f(x)$ detailed in Eq. (5):

$$f\left(x\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}. \quad (5)$$

In practice we assume that each of the probability function has a Gaussian distribution, we then only need to calculate mean μ and variance σ^2 to obtain the density Eq. (5).

We then consider Eq. (6):

$$P\left(x_k|w_i\right) = f(x_k). \quad (6)$$

This classifier is called Gaussian Naïve Bayes. In which we need to compute the mean μ_i and variance σ^2 of each training samples of classes w_i.

For example, in case of two classes, the new observation x is predicted to belong to the class w_1, if $q_1 f_1(x) > q_2 f_2(x)$.

2.3. The Adaboost Algorithm

Adaboost is an algorithm that combine weak classifier and inaccurate rules to get a highly accurate prediction rule. On every iteration, the algorithm focuses on mistakes made by inaccurate rules by adding a notion of weight [13]. The combination of all the rules then makes a more precise one. It can be illustrated as in Fig. 1

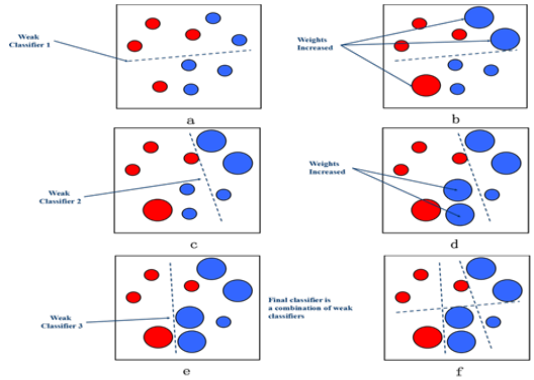

Fig. 1: Adaboost principle.

3. The Boosted Gaussian Naïve Bayes Classifier

The Boosted Gaussian Naïve Bayes Algorithm combines Adaboost and the Gaussian Naïve Bayes classifier. The algorithm classifies a dataset using the Gaussian Naïve Bayes classifier. It then begins a boosting process by adding weight: D_t, to the misclassified samples. The next iteration of Gaussian Naïve Bayes will then focus on the specific misclassified samples. To ensure that the weight added to the misclassified samples is taken into account for the calculation of the final classifier. For every iteration, the weighted error is calculated by Eq. (7):

$$\varepsilon_t = \sum_{i:f_t(x_i)\neq y_i} D^{(t)}(i). \quad (7)$$

This error will be used to calculate a parameter α which will represent the contribution of each hypothesis: h_t, to the final prediction.

The principle of the Boosted Gaussian Naïve Bayes can be translated into this pseudo-algorithm, Fig. 2.

Input :

- A training set S = $((x_1,y_1),...,(x_m,y_m))$.

Initialization :

- Maximum number of iterations T;
- Initialize t at 1;
- Initialize the weight distribution $\forall i \in \{1,...,m\}, D^{(1)}(i) = \frac{1}{m}$.

while $(t < T$ **and** $\varepsilon < 0.5)$ **do**

- Estimate the puff of two classes: $f_1(x)$ and $f_2(x)$
- Learn a classifier $h_t : \sim^d \rightarrow \{-1,+1\}$ with $h_t(x) = f_1(x) - f_2(x)$
- Set $\varepsilon_t = \sum_{i:f_t(x_i) \neq y_i} D^{(t)}(i)$
- Choose $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$
- Update the weight distribution over examples

$$D^{(t+1)}(i) = \frac{D^{(t)}(i) e^{-\alpha_t y_i f_t(x_i)}}{Z^{(t)}}, \forall i \in \{1,...,m\}, \quad (*)$$

where $Z^{(t)} = \sum_{i=1}^{m} D^{(t)}(i) e^{-\alpha_t y_i f_t(x_i)}$ is a normalization factor such that $D^{(t+1)}$ remains a distribution

t = t+1

Output: The voted classifier

$$\forall x, H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Fig. 2: Boosted Gaussian Naïve Bayes Algorithm.

4. Numerical example

In this section, firstly, we explain concretely the bank credit scoring purposes, then three numerical examples, one simulated and two real-life datasets, are carried out to compare the performance of the proposed approach and Gaussian Naïve Bayes.

Our datasets are data from Can Tho city and Vinh Long province banks. The financial market of Viet Nam has strong growth. The banks may have many opportunities and challenges. Our goal is to classify clients using information such as payment interest, length of time using credit, amount of debt a client has and the types of debt that a client has. We classify into two classes, in order to lead the decision to extend or deny credit for example.

In bank credit operations, the important question is how to determine the repayment ability and creditworthiness of a customer. Lenders use a credit scoring system, or a numerical system, to measure how likely it is that a borrower will make payments on the money he or she borrows and to decide on whether to extend or deny credit. Lenders use machine learning algorithm to determine how much risk a particular borrower places on them if they decide to lend to that person. Therefore, the study on assessing the ability to repay bank debt is necessary.

4.1. Example 1 Simulated data

We test the algorithm on a sample of simulated data in order to obtain a training model. We generate 100 random samples according to the following formula Eqs.(8)-(9).

$$w_1 = \left\{ \begin{matrix} (\sqrt{x}\cos(2\pi x), \sqrt{x} sin(2\pi x)) \\ |x \in X, X \sim U(0,1) \end{matrix} \right\} \quad (8)$$

and $w_2 =$

$$\left\{ \begin{matrix} (\sqrt{3x+1}\cos(2\pi x), \sqrt{3x+1} sin(2\pi x)) \\ |x \in X, X \sim U(0,1) \end{matrix} \right\}. \quad (9)$$

We then try to create a model using the Boosted Gaussian Naïve Bayes algorithm. Red points represent instances which belong to w_1 class, green points to w_2 class and blue points are the misclassified one, Fig. 3. After the first classification we can see that 5 samples were misclassified, Fig. 3. We calculate the error, update the weight and make another classification. The algorithm will now focus on the "weak" sample, Fig. 4. After 10 iterations, we combine the classifier, Fig. 5. We obtain the final model:

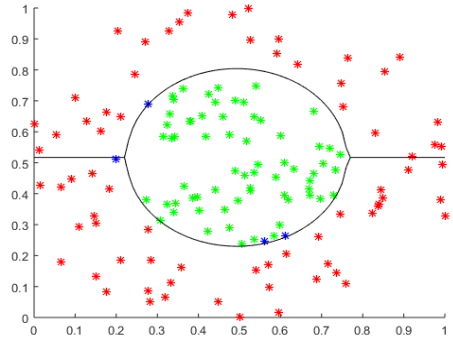

Fig. 3: First Classification using Gaussian Naïve Bayes.

Table 1. Comparison of the results of the two methods.

	Boosted Gaussian Naïve Bayes	Gaussian Naïve Bayes Classifier
Mean error	0.0228	0.0446
Computational Time (s)	4.21	1.22

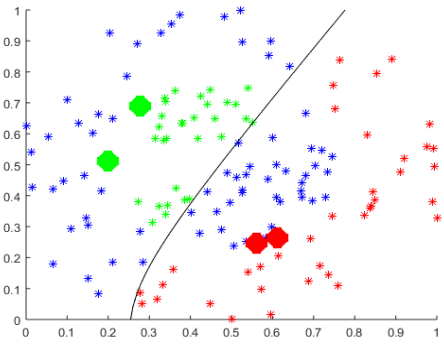

Fig. 4: Classification after the first Boost.

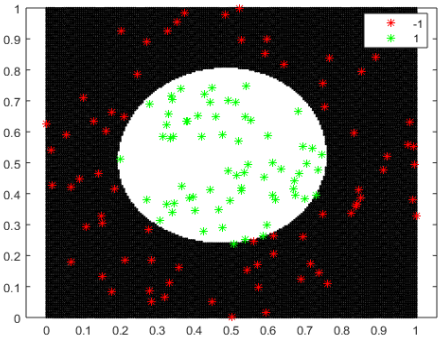

Fig. 5: Final Model.

After 25 iterations of Boosted Gaussian Naïve Bayes versus the Gaussian Naïve Bayes. We obtain the final mean error and the computational time (see Table 1).

The Boosted algorithm provides better results: it is two times more precise than the Gaussian Naïve Bayes classifier. Though, the computational time to create the model is 3.5 times more important.

Thanks to this example, we can identify more precisely the pros and the cons of the Boosted method. The gain in precision is proportional to the complexity increase. In this case an increase of 0.02% is not interesting, in comparison of the augmentation of the computational time.

4.2. Bank in Can Tho city

Our first sample to experiment is a dataset of 71 cases of bad debt and 143 cases of good debt of a bank in Can Tho city. The statistical unit is bank borrowers who are enterprises in strategic sectors, such as agriculture, commerce and industry. 13 independent variables are available in the sample to determine the quality of bank's borrowing. However, to perform the classification, we use only two decisive variables in our experimentation, $X1$ and $X4$, according to testing results only these two variables have statistical significance at the 5% level. $X1$ and $X4$ are correspondingly the financial leverage and the interest of borrowers.

We run the training process 10 times, the error is calculated by averaging error of 10 times and we choose to randomly divide 10 times our dataset into 70% training and 30% test sets in order to obtain reliable results (see Table 2). The Boosted Gaussian Bayes classifier presents a better accuracy than Gaussian Naïve Bayes classifier: $0.273 < 0.317$. Once again, the computational time is around 3.5 times longer. Nevertheless, in this example, the error gain is more interesting. Indeed, the results present that 6% more of the customers are predicted in the correct class. Regarding the augmentation of computational time, a 6% precision increase is relevant especially in the case of a bank loan profit.

4.3. Bank in Vinh Long province

Our second sample to experiment is a dataset about the repayment ability of 166 companies of 24 cases of bad debt and 141 cases of good debt in Vinh Long province. We have three independent variables in our sample (see Table 3). We

Table 2. Comparison of the results.

Boosted Gaussian Bayes error	Gaussian Naïve Bayes error
0.218	0.234
0.296	0.265
0.312	0.312
0.203	0.203
0.265	0.593
0.296	0.312
0.171	0.187
0.218	0.406
0.296	0.390
0.296	0.281
Mean	Mean
0.257	0.318
Computational Time(s)	Computational Time(s)
1.403	0.425

Table 3. Variables of the second sample.

X_i	Detail	Independent variables
X_1	Years in business activity	Management experience
X_2	Total debt/total equity	Financial leverage
X_3	Net sales/Average Total Assets	Asset turnover

Table 4. Comparison result of two methods.

Boosted Gaussian Bayes error	Gaussian Naïve Bayes error
0.120	0.400
0.140	0.400
0.120	0.420
0.160	0.420
0.140	0.460
0.140	0.320
0.140	0.320
0.120	0.580
0.120	0.480
0.120	0.460
Mean	Mean
0.132	0.426
Computational Time(s)	Computational Time(s)
1.443	0.349

use the same process that the subsection above, we run the training process 10 times, the error is calculated by averaging error of 10 times and we choose to randomly divide 10 times our dataset into 70% training and 30% test sets in order to obtain reliable results (see Table 4).
The Boosted Gaussian Bayes classifier presents a better accuracy than Gaussian Naïve Bayes classifier: $0.132 < 0.426$. In this final case, the computational time is around four times more important for the Boosted method, though the gain in precision is around four times more precise. In this case of repayment abilities for the Vinh Long province bank, the use of Boosted algorithm presents an important interest. Around 30% of their customer would be misclassified by using a standard Bayesian classifier. This number is too high to be neglected and will represent a huge loss of money for the bank.

5. CONCLUSION

This article has proposed a pseudo-algorithm aimed at solving classification problems with Boosted Gaussian Naïve Bayes classifier. We manage to overcome its limit by adapting our classifier to misclassified observations. According to the results of each data sample, we can state that Boosted Gaussian Naïve Bayes method is better than Naïve Bayes classifier in classification accuracy, although, the Boosted algorithm requires a huge computational time. In our tests, around 4 times longer than the classic Bayesian model.

The algorithm proposed in this paper presents a great interest in the case of small datasets, the gain in precision is very important and can present, for institutions like banks, an important gain in benefits. For very big datasets, the algorithm computational time will not be interesting enough to justify the gain in precision. This algorithm presents an alternative to Bayesian classifier. One approach to improve this algorithm is to adapt the algorithm to the size of the dataset. The next step for this classification algorithm would be to make it choose between the methods according to the different characteristics of the datasets.

This approach has only been tested in the bank sector. It would be interesting to test this algorithm in other sectors such as medical or agronomic domains to define if the benefits in precision gain would be as interesting as in the bank domain.

References

[1] Elkan, C. (1997). Boosting and Naïve Bayesian learning. In Proceedings of the International Conference on Knowledge Discovery and Data Mining.

[2] Ridgeway, G., Madigan, D., Richardson, T., & O'Kane, J. (1998). Interpretable Boosted Naïve Bayes Classification. In KDD, 101-104.

[3] Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29(2-3), 103-130.

[4] Mitchell, T. M. (1997). Machine learning. WCB.

[5] Hellerstein, J. L., Jayram, T. S., & Rish, I. (2000). Recognizing end-user transactions in performance management. Hawthorne, NY: IBM Thomas J. Watson Research Division.

[6] Nguyen-Trang, T., & Vo-Van, T. (2017). A new approach for determining the prior probabilities in the classification problem by Bayesian method. Advances in Data Analysis and Classification, 11(3), 629-643.

[7] Vo-Van, T., Nguyen-Trang, T., & Ha, C. N. (2016). The prior probability in classifying two populations by Bayesian method. Applied Mathematics Engineering and Reliability, 6, 35-40.

[8] Hilden, J. (1984). Statistical diagnosis based on conditional independence does not require it. Computers in biology and medicine, 14(4), 429-435.

[9] Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. In Aaai (Vol. 90, pp. 223-228).

[10] Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. Machine learning, 29(2-3), 131-163.

[11] Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29(2-3), 103-130.

[12] Vo-Van, T. (2017). Classifying by Bayesian Method and Some Applications. In Bayesian Inference. InTech.

[13] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139.

[14] Schwenk, H., & Bengio, Y. (1998). Training methods for adaptive boosting of neural networks. In Advances in neural information processing systems (pp. 647-653).

[15] Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning, 36(1-2), 105-139.

[16] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine learning, 40(2), 139-157.

About Authors

Pizzo ANAÏS Student from a French engineering school, Polytech Lille, in statistics and computer sciences department for five years. Currently, at Ton Duc Thang University, Ho Chi Minh City, Viet Nam, for two months of internship.

Teyssere PASCAL Student from the French engineering school: Polytech Lille, in internship for 2 months in Vietnam. I have been working on this thesis subject with Anaïs and supervised by Thao Trang Nguyen.