

A Novel Mixed-Attribute Fusion-Based Naive Bayesian Classifier

Guiliang Ou¹, Yulin He^{1,2,*} , Philippe Fournier-Viger^{1,2} and Joshua Zhexue Huang^{1,2}

¹ Big Data Institute, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

² Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518107, China

* Correspondence: yulinhe@gml.ac.cn; Tel.: +86-185-3131-5747

Abstract: The Naive Bayesian classifier (NBC) is a well-known classification model that has a simple structure, low training complexity, excellent scalability, and good classification performances. However, the NBC has two key limitations: (1) it is built upon the strong assumption that condition attributes are independent, which often does not hold in real-life, and (2) the NBC does not handle continuous attributes well. To overcome these limitations, this paper presents a novel approach for NBC construction, called mixed-attribute fusion-based NBC (MAF-NBC). It alleviates the two aforementioned limitations by relying on a mixed-attribute fusion mechanism with an improved autoencoder neural network for NBC construction. MAF-NBC transforms the original mixed attributes of a data set into a series of encoded attributes with maximum independence as a pre-processing step. To guarantee the generation of useful encoded attributes, an efficient objective function is designed to optimize the weights of the autoencoder neural network by considering both the encoding error and the attribute's dependence. A series of persuasive experiments was conducted to validate the feasibility, rationality, and effectiveness of the designed MAF-NBC approach. Results demonstrate that MAF-NBC has superior classification performance than eight state-of-the-art Bayesian algorithms, namely the discretization-based NBC (Dis-NBC), flexible naive Bayes (FNB), tree-augmented naive (TAN) Bayes, averaged one-dependent estimator (AODE), hidden naive Bayes (HNB), deep feature weighting for NBC (DFW-NBC), correlation-based feature weighting filter for NBC (CFW-NBC), and independent component analysis-based NBC (ICA-NBC).

Keywords: naive Bayesian classifier; attribute independence assumption; mixed-attribute classification; conditional probability; Bayesian network; attribute transformation



Citation: Ou, G.; He, Y.; Fournier-Viger, P.; Huang, J.Z. A Novel Mixed-Attribute Fusion-Based Naive Bayesian Classifier. *Appl. Sci.* **2022**, *12*, 10443. <https://doi.org/10.3390/app122010443>

Academic Editor: Xinglong Zhang, Pengfei Jia and Yue Wu

Received: 26 September 2022

Accepted: 10 October 2022

Published: 17 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As one of the top 10 algorithms in the fields of data mining and machine learning [1], the naive Bayesian classifier (NBC) has been used in numerous domains. The main advantage of the NBC is its simple model structure that makes it easy to implement and its good theoretical interpretability. In recent years, the NBC also received much attention from the industry and academia since it can be easily deployed in distributed environments to process big data. Despite possessing several desirable properties, the NBC is built upon a strong assumption, called the attribute independence assumption, which states that condition attributes must be mutually independent with respect to the decision attribute. This assumption simplifies the calculation of posterior probabilities (the probabilities that attribute values of a sample are all observed for a given class). Rather than computing a posterior probability as a joint probability of condition attributes, the NBC calculates it as the product of multiple marginal probabilities. This makes the computation of NBC very efficient and allows the estimation of probabilities even with small data sets. However, the attribute independence assumption does not hold in many real-life data sets, which substantially limits the prediction performance of the NBC. Recent studies on the NBC mainly focused on finding ways to relax the independence assumption so as to further improve its generalization performance. There are two main approaches for improving the

testing ability of the NBC [2]: (1) improving the structure of models and (2) performing data transformations. The former approach introduces complex Bayesian network structures to model attribute dependencies, while the latter applies feature selection or feature extraction methods to select independent attributes. Some representative studies of these two approaches are described next.

Model structure-oriented improvement methods use different estimation strategies (e.g., density estimation, Bayesian network, and attribute weighting) to estimate the conditional probability that a sample's attribute values are observed given its class. The flexible naive Bayes (FNB) [3] relies on kernel density estimations to determine a class's conditional probabilities. However, FNB can only handle continuous attributes and does not offer a solution to relax the attribute independence assumption. The tree-augmented naive (TAN) Bayes [4] algorithm is a semi-naïve Bayesian learning method that relaxes the attribute independence assumption by employing a tree structure, where each condition attribute only depends on the decision attribute and at most one condition attribute. The averaged one-dependent estimator (AODE) [5] is an ensemble classifier that applies an attribute selection algorithm to construct a series of one-dependent classifiers. Each classifier is a simple Bayesian network that is obtained by averaging all one-dependence classifiers. The hidden naive Bayes (HNB) classifier [6] uses the weighted sum of two-attribute dependencies to represent multiple-attribute dependence, where the weights are determined based on the mutual information between two condition attributes. The deep feature weighting for NBC (DFW-NBC) [7] technique and correlation-based feature weighting filter for NBC (CFW-NBC) [8] are two attribute weighting-based NBCs. DFW-NBC estimates each conditional probability of the NBC by deeply computing attribute weighted frequencies from training data while CFW-NBC weights the condition attributes by considering both the mutual relevance and the average mutual redundancy.

Data transformation-oriented improvement methods focus on the deep exploration of the training data to avoid complex and time-consuming structure learning. The NBCs obtained by this approach still rely on the independence assumption; thus, the performance may be degraded if the training data do not satisfy that assumption. Feature extraction techniques are the most commonly used to transform data with attribute dependencies into data that have no dependencies. Bressan and Vitria [9] applied class-conditional independent component analysis (CC-ICA) to improve the classification performance of the NBC. Qin et al. [10] proposed an ICA-based NBC (ICA-NBC) to improve the classification performance of the NBC. Fan and Poh [11] evaluated the classification performance of the NBC built using three feature extraction methods, namely principal component analysis (PCA), ICA, and CC-ICA. Experimental results have shown that PCA, ICA, and CC-ICA can slightly increase the testing accuracies of the NBC on the selected data sets. Jayanthi and Sasikala [12] trained an NBC based on website attributes extracted by PCA to perform web link spam detection. Zhang et al. [13] applied PCA to extract key attributes in network data and then trained the NBC to conduct network intrusion detection.

Although the aforementioned methods can improve the classification performance of the NBC for specific application scenarios, these methods do not provide a good solution for handling mixed attributes (continuous and categorical attributes). Some studies [14,15] revealed that the discretization of continuous attributes can lead to losing precious information about the original data set, while the determination of an optimal Bayesian network structure is an NP-hard problem [16]. To address the above limitations, this paper proposes a new strategy to enhance the generalization capability of the NBC for the mixed-attribute classification problem. The proposed approach not only retains as much information as possible about the original data but also keeps the model structure as simple as possible. The contributions of this paper are summarized as follows. A new mixed attribute fusion-based NBC (MAF-NBC) is proposed for mixed-attribute classification problems. To relax the attribute independence assumption, an autoencoder neural network (ANN) based on the minimization of both encoding error and attribute dependence is iteratively trained to transform the original mixed attributes into a series of independent and encoded at-

tributes. Extensive experiments conducted on multiple data sets demonstrate the feasibility, rationality, and effectiveness of the MAF-NBC.

The remainder of this paper is organized as follows. Section 2 reviews the principles of the NBC. Section 3 presents the proposed MAF-NBC. Section 4 reports results from experiments to assess the performance of MAF-NBC. Finally, Section 5 concludes this paper and describes future studies.

2. Naive Bayesian Classifier for Mixed-Attribute Classification

Let there be a classification data set $\mathbb{D} = \bigcup_{m=1}^{\mathcal{M}} \mathbb{D}^{(m)}$ containing $\mathcal{N}$ samples. Each sample has $\mathcal{D}$ condition attributes including $\mathcal{D}_1$ discrete attributes and $\mathcal{D}_2$ continuous attributes. Moreover, all samples are divided into $\mathcal{M}$ different classes, where $\mathcal{N}_m$ denotes the number of samples that belong to the m -th ($m = 1, 2, \dots, \mathcal{M}$) class.

$$\mathbb{D}^{(m)} = \left\{ \left(\mathbf{x}_n^{(m)}, y_n^{(m)} \right) \middle| \mathbf{x}_n^{(m)} = \left(a_{n1}^{(m)}, \dots, a_{n\mathcal{D}_1}^{(m)}, b_{n1}^{(m)}, \dots, b_{n\mathcal{D}_2}^{(m)} \right), y_n^{(m)} = w_m, n = 1, 2, \dots, \mathcal{N}_m \right\}, \quad (1)$$

$\sum_{m=1}^{\mathcal{M}} \mathcal{N}_m = \mathcal{N}$, $\mathcal{D}_1 + \mathcal{D}_2 = \mathcal{D}$, $a_{ni}^{(m)}$, $i \in \{1, 2, \dots, \mathcal{D}_1\}$ is the i -th discrete attribute value, $b_{nj}^{(m)}$, $j \in \{1, 2, \dots, \mathcal{D}_2\}$ is the j -th continuous attribute value, and $\{w_1, w_2, \dots, w_{\mathcal{M}}\}$ is the class label set. The next paragraphs explain the principles of the naive Bayesian classifier (NBC) for the mixed-attribute classification problem on data set $\mathbb{D}$.

Assume that there is a new sample $\mathbf{x} = (a_1, \dots, a_{\mathcal{D}_1}, b_1, \dots, b_{\mathcal{D}_2})$. The NBC determines its class label by using this equation:

$$\begin{aligned} y &= \arg \max_{m=1,2,\dots,\mathcal{M}} P(w_m | \mathbf{x}) \\ &= \arg \max_{m=1,2,\dots,\mathcal{M}} \frac{P(\mathbf{x} | w_m) P(w_m)}{P(\mathbf{x})}, \\ &\propto \arg \max_{m=1,2,\dots,\mathcal{M}} P(\mathbf{x} | w_m) P(w_m) \end{aligned} \quad (2)$$

where $P(w_m | \mathbf{x})$ is the posterior probability, $P(w_m)$ is the prior probability, and $P(\mathbf{x} | w_m)$ is the conditional probability. Generally, the prior probability in Equation (1) can be calculated as follows.

$$P(w_m) = \frac{\mathcal{N}_m}{\mathcal{N}}. \quad (3)$$

The key of training an NBC for the mixed-attribute classification problem is to calculate the conditional probability based on the independent attribute assumption as follows.

$$\begin{aligned} P(\mathbf{x} | w_m) &= P(a_1, \dots, a_{\mathcal{D}_1}, b_1, \dots, b_{\mathcal{D}_2} | w_m) \\ &= \left[\prod_{i=1}^{\mathcal{D}_1} P(a_i | w_m) \right] \left[\prod_{j=1}^{\mathcal{D}_2} P(b_j | w_m) \right]. \end{aligned} \quad (4)$$

Conditional probability $P(a_i | w_m)$ corresponding to a discrete attribute value a_i is calculated as follows:

$$P(a_i | w_m) = \frac{\sum_{n=1}^{\mathcal{N}_m} I(a_i, a_{ni}^{(m)})}{\mathcal{N}_m}, \quad (5)$$

where

$$I(u, v) = \begin{cases} 1, & \text{if } u = v \\ 0, & \text{if } u \neq v \end{cases} \quad (6)$$

is an indicator function that is used to count the frequency of a_i in the i -th continuous attribute values $a_{1i}^{(m)}, a_{2i}^{(m)}, \dots, a_{N_{m,j}}^{(m)}$ corresponding to samples belonging to the m -th class. John and Langley [3] constructed a flexible NBC (FNBC), which used the kernel density estimation technique [17] to calculate the term of $P(b_j|w_m)$ for the continuous attribute value in Equation (3) as follows:

$$P(b_j|w_m) \propto p(b_j|w_m) = \frac{1}{N_m} \sum_{n=1}^{N_m} \frac{1}{\sqrt{2\pi}h_j} \exp \left[-\frac{1}{2} \left(\frac{b_j - b_{nj}^{(m)}}{h_j} \right)^2 \right] \quad (7)$$

where $p(b_j|w_m)$ is the estimated probability density function (*p.d.f.*) value of b_j based on the j -th continuous attribute values $b_{1j}^{(m)}, b_{2j}^{(m)}, \dots, b_{N_{m,j}}^{(m)}$ corresponding to samples from the m -th class, and $h_j > 0$ ($h_j = \frac{1}{\sqrt{N_m}}$ in [3]) is the bandwidth parameter. In addition, continuous attribute discretization can also be used to determine $p(b_j|w_m)$ as follows:

$$P(b_j|w_m) = P(c_j|w_m) = \frac{\sum_{n=1}^{N_m} I(c_j, c_{nj}^{(m)})}{N_m} \quad (8)$$

by transforming the continuous attribute values $b_j, b_{1j}^{(m)}, b_{2j}^{(m)}, \dots, b_{N_{m,j}}^{(m)}$ into the discrete attribute values $c_j, c_{1j}^{(m)}, c_{2j}^{(m)}, \dots, c_{N_{m,j}}^{(m)}$. This form of discretization-based NBC is called dis-NBC in this study.

3. Mixed-Attribute Fusion-Based Naive Bayesian Classifier

As stated in the introduction, continuous attribute discretization and the attribute independence assumption limit the generalization performance of the NBC for the continuous attribute classification problem. To cope with these issues, recent studies either introduced complex structures to represent attribute dependencies or applied discretization techniques to transform mixed-attribute values into discrete attribute values [18–20]. This section presents a novel solution to the NBC-based mixed-attribute classification problem by considering attribute dependence and mixed-attribute transformation simultaneously. A mixed-attribute fusion strategy is designed to construct an NBC that can be trained based on the transformed continuous attributes with the minimum attribute dependence.

For discrete attributes of an original mixed-attribute (OMA) data set $\mathbb{D}$, the one-hot encoding technique [21] is applied to transform them into 0-1 numerical attributes; i.e., $a_{ni}^{(m)}, i = 1, 2, \dots, D_1$ is encoded as $(e_{n1}^{(mi)}, \dots, e_{nK_i}^{(mi)})$ for the discrete attribute value of the n -th sample $x_n^{(mi)}, n = 1, 2, \dots, N_m$, where the following is the case:

$$e_{nk}^{(m)} = \begin{cases} 1, & \text{if } a_{ni}^{(m)} = A_k^{(i)} \\ 0, & \text{if } a_{ni}^{(m)} \neq A_k^{(i)} \end{cases} \quad (9)$$

and $\{A_1^{(i)}, A_2^{(i)}, \dots, A_{K_i}^{(i)}\}$ is K_i categorical values corresponding to the i -th discrete attribute. Then, the one-hot encoded form of the original sample $x_n^{(m)}$ can be expressed as follows.

$$\bar{x}_n^{(m)} = (e_{n1}^{(m1)}, \dots, e_{nK_1}^{(m1)}, \dots, e_{n1}^{(mD_1)}, \dots, e_{nK_{D_1}}^{(mD_1)}, b_{n1}^{(m)}, \dots, b_{nD_2}^{(m)}) \quad (10)$$

In fact, the one-hot encoding technique causes attribute redundancy when extending a discrete attribute into multiple 0-1 numerical attributes. For example, the essence of the following transformation

$$\begin{bmatrix} A_1 \\ A_1 \\ A_2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (11)$$

is to represent a discrete attribute using two “repetitive” numerical attributes. The main role of one-hot encoding is to transform the discrete attribute values into numbers. The attribute independence assumption is not alleviated when constructing an NBC based on a one-hot encoded attribute (OHEA) data set.

$$\bar{X} = \begin{bmatrix} \bar{x}_1^{(1)} & \cdots & \bar{x}_{N_1}^{(1)} & \bar{x}_1^{(2)} & \cdots & \bar{x}_{N_2}^{(2)} & \cdots & \bar{x}_1^{(\mathcal{M})} & \cdots & \bar{x}_{N_M}^{(\mathcal{M})} \end{bmatrix}^T. \quad (12)$$

Thus, an autoencoder neural network (ANN) is meticulously designed to solve the problems of attribute redundancy and attribute dependence mentioned above.

An ANN is a special single hidden-layer feed-forward neural network that has the same input matrix and output matrix. The main purpose of training an ANN is to find the optimal input layer weight matrix $\alpha = (\alpha_{rl})_{\mathcal{R} \times \mathcal{L}} = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_{\mathcal{L}}]$ and output layer matrix $\beta = (\beta_{lr})_{\mathcal{L} \times \mathcal{R}}$ so that the practical output matrix $\bar{X}'$ approximates the true output matrix $\bar{X}$ as closely as possible, where $\mathcal{R} = \sum_{i=1}^{\mathcal{D}_1} \mathcal{K}_i + \mathcal{D}_2$ is the number of input layer nodes of the ANN and $\mathcal{L}$ is the number of hidden layer nodes of the ANN.

The following objective function for ANN is designed to transform one-hot encoded attributes into independent encoded attributes:

$$L(\bar{X}, \alpha, \beta) = \lambda L_1(\bar{X}, \alpha, \beta) + (1 - \lambda) L_2(\bar{X}, \alpha, \beta), \quad (13)$$

where $\lambda \in (0, 1)$ is a balance factor. Optimal weight matrices $\bar{\alpha}$ and $\bar{\beta}$ are determined by minimizing the objective function as follows:

$$\bar{\alpha}, \bar{\beta} = \underset{\substack{\alpha_{rl}, \beta_{lr} \in \mathbb{R} \\ r=1,2,\dots,\mathcal{R}; l=1,2,\dots,\mathcal{L}}}{\operatorname{argmax}} \{L(\bar{X}, \alpha, \beta)\}. \quad (14)$$

Here, it is worthwhile to note that the original intention of using an autoencoder neural network is to fuse the mixed attributes and to relax the attribute independence assumption rather than exploring the usage of deep learning technology. When a shallow learning can already meet the requirement of good NBC construction, it is unnecessary to resort to complex and time-consuming deep learning. The excessive attention on attribute transformation with deep learning is out of the scope of this study. Interested readers can refer to specialized studies on the combination of deep learning and supervised learning for more details on such approaches, e.g., deep support vector machine [22], deep decision tree [23], and deep nearest neighbor [24].

The first term of the objective function is the encoding error, which is used to measure the error between the practical output matrix $\bar{X}'$ and the true output matrix $\bar{X}$. It is defined as follows:

$$L_1(\bar{X}, \alpha, \beta) = \|\bar{X}' - \bar{X}\|_2^2 = \|\operatorname{sigmoid}(\bar{X}\alpha)\beta - \bar{X}\|_2^2 \quad (15)$$

where

$$\bar{X} = \operatorname{sigmoid}(\bar{X}\alpha) = \begin{bmatrix} \bar{x}_1^{(1)} & \cdots & \bar{x}_{N_1}^{(1)} & \cdots & \bar{x}_1^{(\mathcal{M})} & \cdots & \bar{x}_{N_M}^{(\mathcal{M})} \end{bmatrix}^T \quad (16)$$

is a $\mathcal{N} \times \mathcal{L}$ hidden-layer output matrix that is a class-independent encoded-attribute (IEA) data set, such that $\bar{x}_n^{(m)} = (\bar{x}_{n1}^{(m)}, \dots, \bar{x}_{n\mathcal{L}}^{(m)}) = (\text{sigmoid}(\bar{x}_n^{(m)} \alpha_1), \dots, \text{sigmoid}(\bar{x}_n^{(m)} \alpha_{\mathcal{L}}))$ is the independent encoded form of the original sample $x_n^{(m)}$ and

$$\text{sigmoid}(s) = \frac{1}{1 + \exp(-s)}, s \in (-\infty, +\infty) \quad (17)$$

is the sigmoid activation function. To minimize attribute dependence, the attribute dependence term in the objective function is defined as follows:

$$L_2(\bar{X}, \alpha, \beta) = \frac{2}{\mathcal{L}(\mathcal{L} - 1)} \sum_{i=1}^{\mathcal{L}} \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{L}} I(h_i, h_j), \quad (18)$$

where

$$h_i = (\bar{x}_{1i}^{(1)}, \dots, \bar{x}_{\mathcal{N}_1 i}^{(1)}, \dots, \bar{x}_{1i}^{(\mathcal{M})}, \dots, \bar{x}_{\mathcal{N}_{\mathcal{M}} i}^{(\mathcal{M})})^T \quad (19)$$

is the i -th independent encoded attribute, and $I(h_i, h_j)$ is the mutual information between independent encoded attributes h_i and h_j and $i, j \in \{1, 2, \dots, \mathcal{L}\}$ and $i \neq j$.

The updating rules of α_{rl} and β_{lr} are derived as follows. The second term of the objective function is unrelated to the output-layer weights. Thus, the updating rule mainly depends on the encoding error. The gradient descent method is used to determine the updating rule of β_{lr} . The partial derivative of $L(\bar{X}, \alpha, \beta)$ with respect to β_{lr} is calculated as follows:

$$\Delta \beta_{lr} = \frac{\partial L(\bar{X}, \alpha, \beta)}{\partial \beta_{lr}} = \lambda \frac{\partial L_1(\bar{X}, \alpha, \beta)}{\partial \beta_{lr}} \quad (20)$$

and then the updating rule of β_{lr} is given as follows

$$\beta_{lr} \leftarrow \beta_{lr} - \xi \times \Delta \beta_{lr}, \quad (21)$$

where $\xi > 0$ is the learning rate. For the input-layer weight α_{rl} , the updating rule cannot be derived by using the gradient descent method because of the existence of mutual information terms. Here, a new updating strategy based on the Monte Carlo method [25] is designed as follows:

$$\alpha_{rl} \leftarrow \alpha_{rl} - \zeta \times \Delta \alpha_{rl}, \quad (22)$$

where $\zeta > 0$ is the learning rate and

$$\Delta \alpha_{rl} = \frac{1}{\mathcal{N}} \sum_{m=1}^{\mathcal{M}} \sum_{n=1}^{\mathcal{N}_m} L(\bar{x}_n^{(m)}, \alpha, \beta) \quad (23)$$

is the approximation of the gradient $\frac{\partial L(\bar{X}, \alpha, \beta)}{\partial \alpha_{rl}}$.

Based on the IEA data set, mixed-attribute fusion-based NBC (MAF-NBC) determines the class label for a given new sample $x = (a_1, \dots, a_{\mathcal{D}_1}, b_1, \dots, b_{\mathcal{D}_2})$ as follows. First, the one-hot encoded form of x is expressed as follows:

$$\bar{x} = (e_1^{(1)}, \dots, e_{\mathcal{K}_1}^{(1)}, \dots, e_1^{(\mathcal{D}_1)}, \dots, e_{\mathcal{K}_{\mathcal{D}_1}}^{(\mathcal{D}_1)}, b_1, \dots, b_{\mathcal{D}_2}), \quad (24)$$

where the following

$$e_k^{(i)} = \begin{cases} 1, & \text{if } a_i = A_k^{(i)} \\ 0, & \text{if } a_i \neq A_k^{(i)} \end{cases}, k = 1, 2, \dots, \mathcal{K}_i, i = 1, 2, \dots, \mathcal{D}_1. \quad (25)$$

Second, $\bar{x}$ is transformed into the following independent encoded expression:

$$\bar{\bar{x}} = (\bar{\bar{x}}_1, \bar{\bar{x}}_2, \dots, \bar{\bar{x}}_{\mathcal{L}}) = (\text{sigmoid}(\bar{x}\alpha_1), \text{sigmoid}(\bar{x}\alpha_2), \dots, \text{sigmoid}(\bar{x}\alpha_{\mathcal{L}})) \quad (26)$$

based on the trained ANN with input weight matrix $\alpha = (\alpha_{rl})_{\mathcal{R} \times \mathcal{L}}$. Third, the class label of x is determined according to Equation (1), where the conditional probability is calculated as follows:

$$\begin{aligned} P(x|w_m) &= P(\bar{x}|w_m) = P(\bar{\bar{x}}|w_m) \\ &= \prod_{l=1}^{\mathcal{L}} P(\bar{\bar{x}}_l|w_m) = \prod_{l=1}^{\mathcal{L}} \int_{-\infty}^{\bar{\bar{x}}_l} p(s|w_m) \end{aligned} \quad (27)$$

where $p(s|w_m)$ is approximated with a normal *p.d.f.*

$$\begin{aligned} f_l^{(m)}(s) &= \frac{1}{\sqrt{2\pi}\sigma_l^{(m)}} \exp \left[-\frac{1}{2} \left(\frac{s - \mu_l^{(m)}}{\sigma_l^{(m)}} \right)^2 \right], \\ s &\in (-\infty, +\infty), m = 1, 2, \dots, \mathcal{M}, l = 1, 2, \dots, \mathcal{L} \end{aligned} \quad (28)$$

with the mean value

$$\mu_l^{(m)} = \frac{1}{\mathcal{N}_m} \sum_{n=1}^{\mathcal{N}_m} \bar{\bar{x}}_{nl}^{(m)} \quad (29)$$

and standard deviation (std).

$$\sigma_l^{(m)} = \sqrt{\frac{1}{\mathcal{N}_m - 1} \sum_{n=1}^{\mathcal{N}_m} [\bar{\bar{x}}_{nl}^{(m)} - \mu_l^{(m)}]^2}. \quad (30)$$

Here, an in-depth discussion regarding the normal *p.d.f.* $f_l^{(m)}(\bullet)$ is given. The sigmoid activation function is used in the designed ANN; thus, the outputs corresponding to each hidden layer's nodes obey a quasi-normal probability distribution. A visual comparison of the sigmoid activation function and normal probability distribution functions with standard deviations 0.1, 0.5, 1.0, and 2.0, as shown in Figure 1, clearly demonstrates this empirical conclusion.

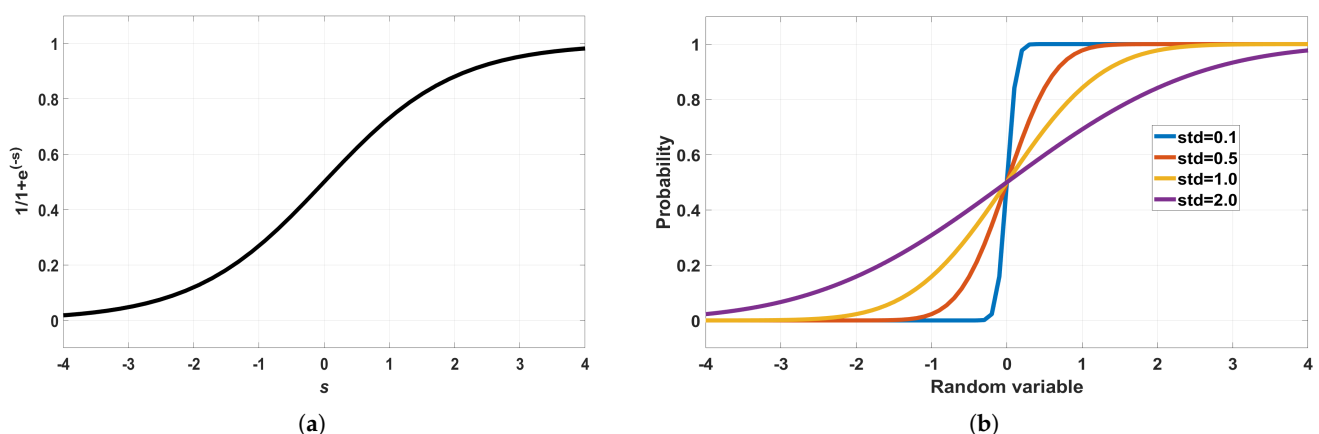


Figure 1. Graphical comparison between sigmoid activation and normal distribution. (a) Sigmoid activation function. (b) Normal probability distribution functions.

For independent attributes, the joint probability distribution is the normal probability distribution if marginal probability distributions are normal probability distributions. Thus, the joint probability $P(\bar{\bar{x}}|w_m)$ can be modeled as the product of multiple marginal probabilities $P(\bar{\bar{x}}_l|w_m)$, $l = 1, 2, \dots, \mathcal{L}$. For the sake of simplicity, hidden-layer biases are not used in the constructed ANN. The role of hidden-layer biases is to control the

bias between the hidden layer's input and the original point of the sigmoid function. When conducting classification or regression tasks, hidden-layer biases are helpful for the generation of high-performance learners. In MAF-NBC, the ANN is designed to transform one-hot encoded attributes into independent encoded attributes that are expected to have minimum dependence. The reasonable objective function shown in Equation (11) is able to guarantee the generation of independent encoded attributes even though hidden-layer biases are not used in ANN. The following experiments will support the aforementioned discussion and conclusion.

4. Experimental Settings and Results

A series of experiments are conducted in this section to validate the feasibility, rationality, and effectiveness of the proposed mixed-attribute fusion-based naive Bayesian classifier (MAF-NBC). Experiments were conducted using 20 KEEL [26] mixed-attribute data sets. Their characteristics are listed in Table 1, where the Arabic numerals in parentheses represent the numbers of available categorical values corresponding to the discrete attributes. The data processing strategy proposed by Helal and Otero [27] was used to generate the mixed attributes for the data sets marked without '*' in Table 1. Then, the equal width discretization method was used to transform the continuous attributes into discrete attributes. All data sets as shown in Table 1 can be downloaded from our BaiDuPan or GitHub online storage space. The experiments were run on a PC equipped with an Intel(R) Quadcore 3.00 GHz i5-9400 CPU and 16 GB of RAM.

Table 1. Details of 20 mixed-attribute data sets.

Data sets	Abalone_3_4	Abalone_9_10	Adult *	Band *
Number of samples	423	1073	5000	540
Number of continuous attributes	6	6	9	18
Number of discrete attributes	2 (2, 4)	2 (2, 4)	5 (5, 7, 5, 4, 2)	5 (3, 5, 2, 3, 3)
Number of classes	2	2	2	2
data sets	Bd	Bp	Heart	Ionosphere
Number of samples	569	198	270	351
Number of continuous attributes	23	25	9	25
Number of discrete attributes	7 (4, 6, 5, 3, 2, 4, 3)	8 (6, 3, 5, 4, 5, 3, 4, 2)	4 (4, 3, 3, 4)	7 (4, 6, 5, 3, 4, 3, 5)
Number of classes	2	2	2	2
Data sets	Libras	Page blocks	Parkinsons	Ring
Number of samples	360	547	195	500
Number of continuous attributes	85	7	17	15
Number of discrete attributes	15 (4, 5, 7, 5, 8, 4, 5, 6, 4, 5, 6, 7, 3, 4, 5)	3 (3, 5, 4)	5 (4, 5, 6, 5, 3)	5 (3, 4, 5, 3, 4)
Number of classes	15	5	2	2
Data sets	Segment	Sonar	SPECTF	Vehicle
Number of samples	2310	208	267	846
Number of continuous attributes	14	50	36	6
Number of discrete attributes	5 (3, 6, 5, 4, 3)	10 (6, 4, 3, 5, 6, 7, 5, 3, 4, 6)	8 (4, 6, 5, 5, 3, 6, 7, 4)	2 (4, 3)
Number of classes	7	2	2	4
Data sets	Vowel	Wine	WineQR	WineQW
Number of samples	528	178	1599	489
Number of continuous attributes	7	10	8	8
Number of discrete attributes	3 (3, 3, 4)	3 (4, 5, 4)	3 (4, 4, 3)	3 (3, 3, 3)
Number of classes	11	3	6	6

4.1. Feasibility Validation of MAF-NBC

A first experiment was performed to validate the feasibility of the MAF-NBC method by checking the convergence of ANN weights by the iterative update process. This experiment was conducted on the representative *Vowel* data set, which has three discrete attributes and seven continuous attributes. The experimental results are the average values corresponding to 10 independent ANN training. Ten ANNs with 50 hidden-layer nodes were constructed with random weights in the $[-1, 1]$ range, a balance factor $\lambda = 0.50$, and learning rates $\xi = \zeta = 0.01$. Figure 2 depicts the variation trends of the 1-norms for the input layer and output layer weights as the iteration number increased. Figure 3

presents the variation trends of encoding error and attribute dependence as the iteration number increased.

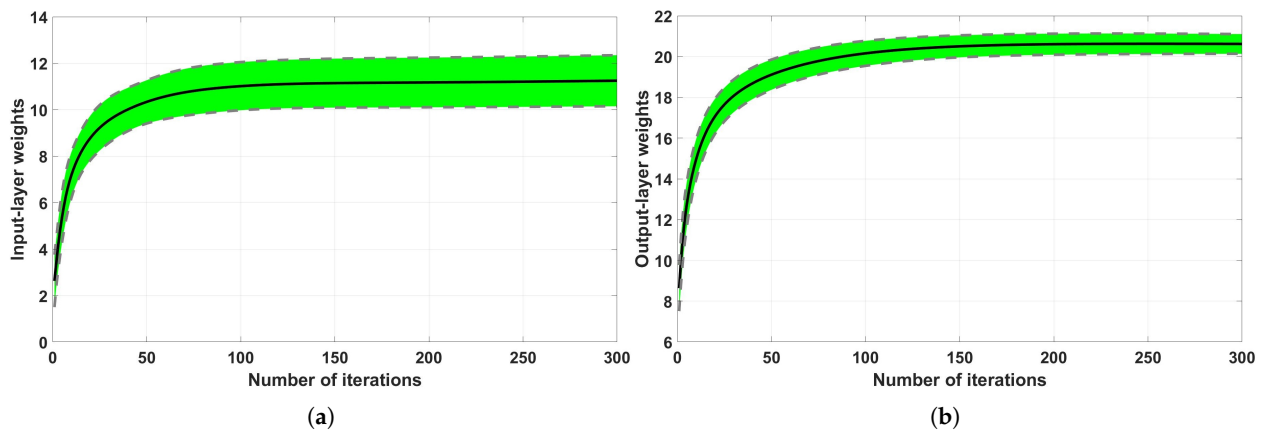


Figure 2. Convergence of the ANN's weights. (a) Input-layer weights. (b) Output-layer weights.

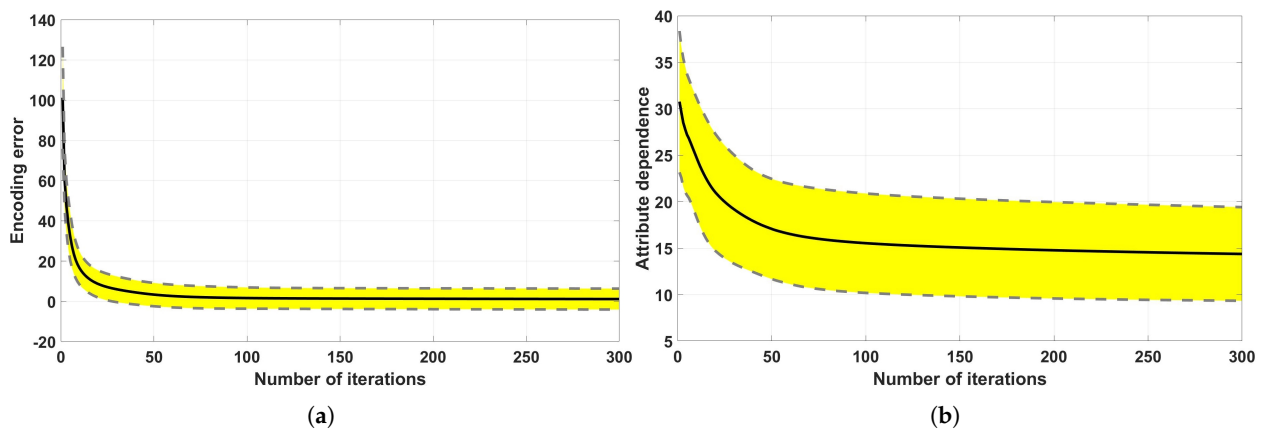


Figure 3. Convergence of the ANN's objective function. (a) Encoding error. (b) Attribute dependence.

Figure 2 shows that the sums of absolute values corresponding to the input layer and output-layer weights become more and more stable with the number of iterations. In addition, Figure 3 shows that the encoding error and attribute dependence decrease gradually and then remain unchanged. These experimental results reveal that the updating rules of Equations (15) and (16) are effective for the determination of optimal ANN weights.

In fact, Equation (17) is the approximation of the expected loss $E_{\mathcal{X}} \left[\frac{\partial L(\bar{\mathcal{X}}, \alpha, \beta)}{\partial \alpha_{rl}} \right]$, where $\bar{\mathcal{X}}$ is the domain of the objective function $L(\bullet, \alpha, \beta)$. Although the gradient of input layer weights cannot be analytically calculated due to the existence of the attribute dependence term, Equation (17) uses the mean of the objective function values corresponding to all samples in the one-hot encoded attribute data set to approximate the gradient. The convergence of output layer weights guarantees the convergence of input layer weights, because the update of α_{rl} depends on the update of β_{lr} . This experiment demonstrates the feasibility of transforming the one-hot encoded attributes into independent encoded attributes.

4.2. Rationality Validation of MAF-NBC

A second experiment was carried out to evaluate the rationality of MAF-NBC in terms of whether the designed ANN can transform the one-hot encoded attributes corresponding to the original mixed attributes into independent encoded attributes. On the representative *Page_small* data set, an ANN with 50 hidden-layer nodes was constructed with random weights in the $[-1, 1]$ interval, a balance factor $\lambda = 0.50$, and learning rates $\zeta = \zeta = 0.01$. Ten representative encoded attributes were selected from the hidden-layer

outputs of ANN. The dependence between the two encoded attributes was measured with the mutual information, which is calculated with *sklearn.metrics.mutual_info_score* (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html, accessed on 14 October 2022) package of the *scikit-learn* machine learning library.

Figure 4 displays a series of heatmaps corresponding to iterations #1, #50, #100, and #150. These heatmaps show the change in attribute dependence as the iteration number increases during the ANN's training. It can be clearly seen that the dependence between encoded attribute gradually decreases as ANN weights are updated. It indicates that independent attributes can be obtained by transforming the original mixed attributes into the encoded attributes. The experimental results shown in Figure 4 are consistent with the experimental results shown in Figure 3b, where the attribute dependence gradually decreases with the increase in iteration number. This experiment demonstrated that the ANN is able to transform the original mixed attributes into independent encoded attributes for NBC construction.

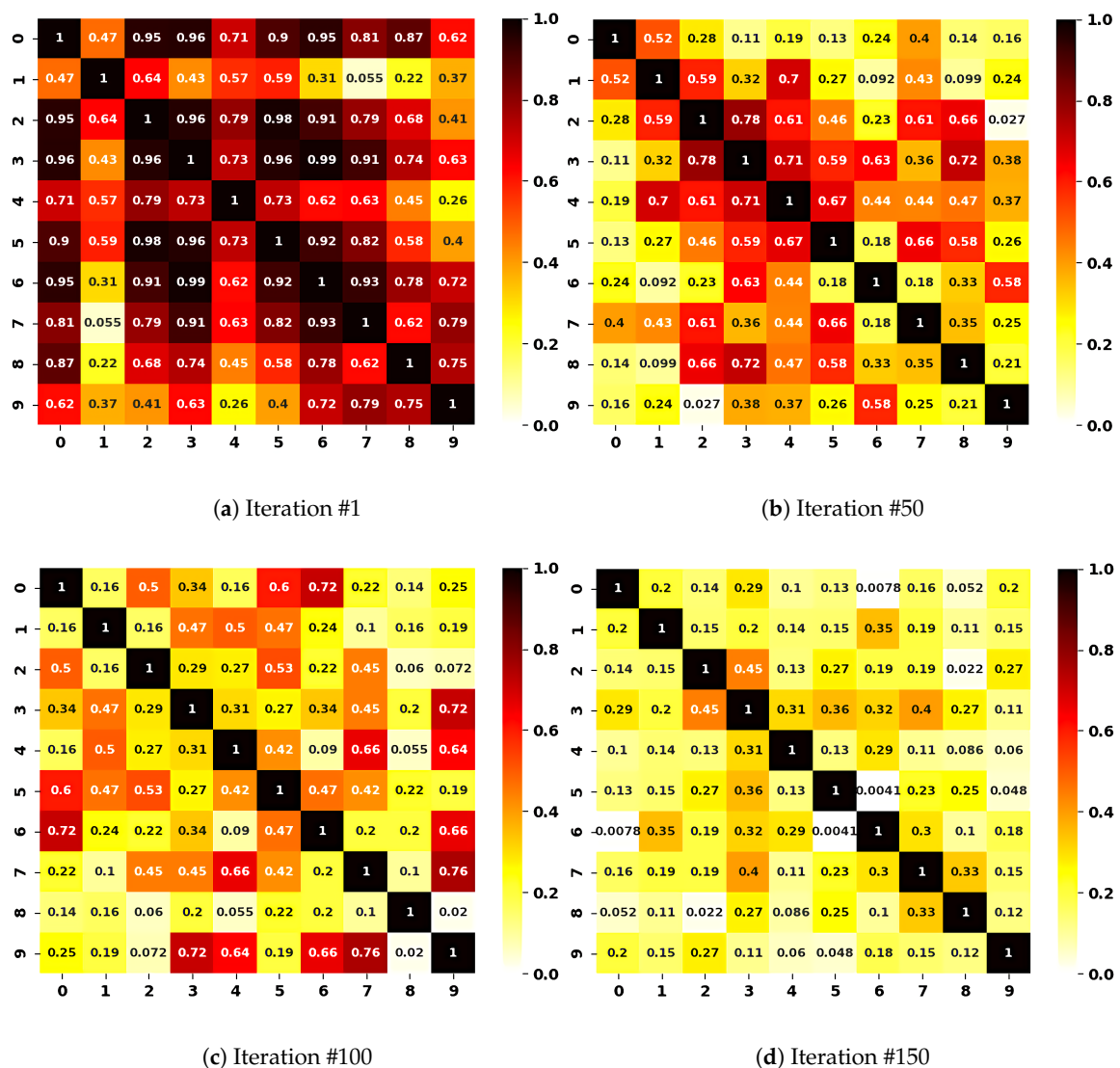


Figure 4. Dependence change of the encoded continuous attributes.

4.3. Effectiveness Validation of MAF-NBC

A third experiment was performed to evaluate the effectiveness of MAF-NBC. This was performed by comparing the classification performances of MAF-NBC with Dis-NBC, FNBC [3], TAN [4], AODE [5], HNB [6], DFW-NBC [7], CFW-NBC [8], and ICA-NBC [10]. All Bayesian algorithms were implemented using the Python programming language. Twenty KEEL mixed-attribute data sets (described in Table 1) were selected to test the training and testing accuracies of the compared algorithms. For each data set, independent training and testing were conducted 10 times with random data set partitions, i.e., 70% samples to train the different Bayes algorithms and 30% to test their generalization capabilities. All ANNs were constructed with 2L hidden-layer nodes, initialized with random weights in the $[-1, 1]$ range, and setup with the learning parameters $\lambda = 0.50$ and $\xi = \zeta = 0.01$. The mean and standard derivation of the 10 training and testing accuracies are listed for each algorithm in Tables 2 and 3, respectively.

The training and testing performances of MAF-NBC were statistically validated by comparing with eight other Bayes algorithms on the 20 data sets. For the given significance level of 0.05, the critical difference (CD) value [28] is calculated as follows:

$$CD = 3.102 \times \sqrt{\frac{9 \times (9 + 1)}{6 \times 20}} \approx 2.686, \quad (31)$$

where the number of compared algorithms is nine, and the number of data sets is 20. In Figure 5, an interval of one CD value can be observed to the left and right of the average rank of MAF-NBC. Any algorithm with a rank outside this area is significantly different from MAF-NBC. It is found that the ranks of MAF-NBC corresponding to the training and testing accuracies are obviously smaller than the other algorithms. In addition, the number of wins for MAF-NBC on 20 data sets is at least $\frac{20}{2} + 1.96 \times \frac{\sqrt{20}}{2} \approx 14$ for the significance level of 0.05. Furthermore, it can be said that MAF-NBC has significantly improved testing accuracies than Dis-NBC (20 wins), FNBC (20 wins), TAN (20 wins), AODE (20 wins), HNB (20 wins), DFW-NBC (19 wins), CFW-NBC (19 wins), and ICA-NBC (18 wins) under a significance level of 0.10. This indicates that MAF-NBC is significantly better for classification than the other algorithms on the selected data sets. The experimental results and statistical analyses demonstrate the effectiveness of MAF-NBC and indicate that MAF-NBC is a viable method to handle mixed-attribute classification tasks. MAF-NBC does not modify the simple model structure of the NBC and preserves the amount of information of the original mixed-attribute data set as much as possible, because the ANN transforms the original mixed attributes into encoded attributes rather than selecting or extracting independent attributes from the original mixed attributes.

Table 2. Comparison of the training accuracies of MAF-NBC, Dis-NBC, FNB, TAN, AODE, HNB, DFW-NBC, CFW-NBC, and ICA-NBC.

Data Set	MAF-NBC	Dis-NBC	FNBC	TAN	AODE	HNB	DFW-NBC	CFW-NBC	ICA-NBC
Abalone_3_4	0.803 $\pm$ 0.036	0.721 $\pm$ 0.032	0.736 $\pm$ 0.026	0.767 $\pm$ 0.038	0.755 $\pm$ 0.032	0.744 $\pm$ 0.029	0.747 $\pm$ 0.018	0.755 $\pm$ 0.017	0.781 $\pm$ 0.045
Abalone_9_10	0.607 $\pm$ 0.029	0.597 $\pm$ 0.012	0.592 $\pm$ 0.021	0.595 $\pm$ 0.025	0.596 $\pm$ 0.024	0.587 $\pm$ 0.041	0.617 $\pm$ 0.008	0.605 $\pm$ 0.014	0.619 $\pm$ 0.017
Adult	0.870 $\pm$ 0.013	0.782 $\pm$ 0.014	0.792 $\pm$ 0.015	0.807 $\pm$ 0.013	0.829 $\pm$ 0.008	0.811 $\pm$ 0.017	0.805 $\pm$ 0.011	0.813 $\pm$ 0.006	0.851 $\pm$ 0.015
Band	0.718 $\pm$ 0.027	0.669 $\pm$ 0.028	0.637 $\pm$ 0.034	0.671 $\pm$ 0.035	0.663 $\pm$ 0.024	0.681 $\pm$ 0.028	0.703 $\pm$ 0.015	0.718 $\pm$ 0.017	0.714 $\pm$ 0.016
Bd	0.991 $\pm$ 0.002	0.919 $\pm$ 0.004	0.938 $\pm$ 0.010	0.961 $\pm$ 0.009	0.959 $\pm$ 0.008	0.953 $\pm$ 0.011	0.959 $\pm$ 0.007	0.970 $\pm$ 0.004	0.983 $\pm$ 0.007
Bp	0.852 $\pm$ 0.022	0.788 $\pm$ 0.023	0.793 $\pm$ 0.027	0.803 $\pm$ 0.029	0.801 $\pm$ 0.034	0.791 $\pm$ 0.041	0.841 $\pm$ 0.036	0.824 $\pm$ 0.033	0.837 $\pm$ 0.027
Heart	0.924 $\pm$ 0.014	0.699 $\pm$ 0.013	0.824 $\pm$ 0.017	0.858 $\pm$ 0.015	0.849 $\pm$ 0.021	0.867 $\pm$ 0.029	0.886 $\pm$ 0.013	0.905 $\pm$ 0.015	0.901 $\pm$ 0.017
Ionosphere	0.981 $\pm$ 0.005	0.901 $\pm$ 0.006	0.917 $\pm$ 0.011	0.934 $\pm$ 0.009	0.932 $\pm$ 0.013	0.926 $\pm$ 0.021	0.928 $\pm$ 0.012	0.941 $\pm$ 0.015	0.974 $\pm$ 0.004
Libras	0.905 $\pm$ 0.007	0.938 $\pm$ 0.015	0.857 $\pm$ 0.019	0.899 $\pm$ 0.015	0.847 $\pm$ 0.016	0.864 $\pm$ 0.015	0.911 $\pm$ 0.016	0.910 $\pm$ 0.015	0.908 $\pm$ 0.011
Page_small	0.952 $\pm$ 0.010	0.915 $\pm$ 0.005	0.911 $\pm$ 0.007	0.909 $\pm$ 0.018	0.919 $\pm$ 0.014	0.914 $\pm$ 0.017	0.931 $\pm$ 0.006	0.933 $\pm$ 0.008	0.937 $\pm$ 0.007
Parkinsons	0.872 $\pm$ 0.029	0.782 $\pm$ 0.021	0.801 $\pm$ 0.022	0.808 $\pm$ 0.025	0.814 $\pm$ 0.027	0.821 $\pm$ 0.021	0.827 $\pm$ 0.022	0.841 $\pm$ 0.025	0.852 $\pm$ 0.024
Ring	0.937 $\pm$ 0.015	0.801 $\pm$ 0.017	0.836 $\pm$ 0.015	0.855 $\pm$ 0.028	0.869 $\pm$ 0.024	0.858 $\pm$ 0.027	0.916 $\pm$ 0.010	0.948 $\pm$ 0.006	0.956 $\pm$ 0.008
Segment	0.940 $\pm$ 0.007	0.878 $\pm$ 0.010	0.889 $\pm$ 0.019	0.897 $\pm$ 0.016	0.895 $\pm$ 0.014	0.901 $\pm$ 0.016	0.928 $\pm$ 0.006	0.945 $\pm$ 0.009	0.959 $\pm$ 0.005
Sonar	0.865 $\pm$ 0.019	0.823 $\pm$ 0.029	0.834 $\pm$ 0.027	0.839 $\pm$ 0.025	0.844 $\pm$ 0.018	0.840 $\pm$ 0.025	0.841 $\pm$ 0.027	0.847 $\pm$ 0.029	0.851 $\pm$ 0.024
Spectf	0.908 $\pm$ 0.011	0.768 $\pm$ 0.023	0.771 $\pm$ 0.017	0.783 $\pm$ 0.019	0.779 $\pm$ 0.015	0.756 $\pm$ 0.029	0.779 $\pm$ 0.016	0.789 $\pm$ 0.014	0.883 $\pm$ 0.019
Vehicle	0.771 $\pm$ 0.016	0.537 $\pm$ 0.005	0.544 $\pm$ 0.011	0.576 $\pm$ 0.016	0.589 $\pm$ 0.019	0.567 $\pm$ 0.026	0.604 $\pm$ 0.022	0.611 $\pm$ 0.021	0.705 $\pm$ 0.017
Vowel	0.869 $\pm$ 0.017	0.829 $\pm$ 0.021	0.839 $\pm$ 0.017	0.836 $\pm$ 0.019	0.840 $\pm$ 0.016	0.847 $\pm$ 0.019	0.843 $\pm$ 0.021	0.848 $\pm$ 0.016	0.833 $\pm$ 0.023
Wine	0.999 $\pm$ 0.001	0.963 $\pm$ 0.021	0.967 $\pm$ 0.017	0.975 $\pm$ 0.006	0.981 $\pm$ 0.004	0.968 $\pm$ 0.007	0.999 $\pm$ 0.001	0.999 $\pm$ 0.001	0.968 $\pm$ 0.019
WineQR	0.738 $\pm$ 0.023	0.563 $\pm$ 0.014	0.578 $\pm$ 0.034	0.580 $\pm$ 0.033	0.591 $\pm$ 0.031	0.602 $\pm$ 0.037	0.638 $\pm$ 0.020	0.717 $\pm$ 0.019	0.688 $\pm$ 0.036
WineQW	0.584 $\pm$ 0.028	0.487 $\pm$ 0.014	0.524 $\pm$ 0.016	0.537 $\pm$ 0.024	0.533 $\pm$ 0.016	0.541 $\pm$ 0.021	0.551 $\pm$ 0.028	0.561 $\pm$ 0.025	0.577 $\pm$ 0.021
Average	0.854 $\pm$ 0.017	0.768 $\pm$ 0.016	0.779 $\pm$ 0.019	0.795 $\pm$ 0.021	0.794 $\pm$ 0.016	0.792 $\pm$ 0.024	0.813 $\pm$ 0.016	0.824 $\pm$ 0.015	0.839 $\pm$ 0.018

Table 3. Comparison of the testing accuracies of MAF-NBC, Dis-NBC, FNB, TAN, AODE, HNB, DFW-NBC, CFW-NBC, and ICA-NBC.

Data Set	MAF-NBC	Dis-NBC	FNBC	TAN	AODE	HNB	DFW-NBC	CFW-NBC	ICA-NBC
Abalone_3_4	0.759 $\pm$ 0.030	0.685 $\pm$ 0.042	0.701 $\pm$ 0.049	0.742 $\pm$ 0.032	0.733 $\pm$ 0.037	0.728 $\pm$ 0.029	0.723 $\pm$ 0.031	0.751 $\pm$ 0.027	0.755 $\pm$ 0.043
Abalone_9_10	0.603 $\pm$ 0.028	0.587 $\pm$ 0.024	0.588 $\pm$ 0.026	0.587 $\pm$ 0.027	0.589 $\pm$ 0.028	0.581 $\pm$ 0.030	0.589 $\pm$ 0.031	0.582 $\pm$ 0.031	0.602 $\pm$ 0.032
Adult	0.842 $\pm$ 0.014	0.795 $\pm$ 0.011	0.788 $\pm$ 0.019	0.801 $\pm$ 0.025	0.817 $\pm$ 0.012	0.807 $\pm$ 0.027	0.804 $\pm$ 0.012	0.811 $\pm$ 0.014	0.824 $\pm$ 0.009
Band	0.647 $\pm$ 0.045	0.606 $\pm$ 0.028	0.601 $\pm$ 0.028	0.626 $\pm$ 0.036	0.618 $\pm$ 0.038	0.609 $\pm$ 0.031	0.619 $\pm$ 0.034	0.630 $\pm$ 0.036	0.619 $\pm$ 0.041
Bd	0.973 $\pm$ 0.011	0.915 $\pm$ 0.017	0.921 $\pm$ 0.013	0.956 $\pm$ 0.012	0.950 $\pm$ 0.010	0.946 $\pm$ 0.014	0.933 $\pm$ 0.015	0.944 $\pm$ 0.016	0.948 $\pm$ 0.014
Bp	0.764 $\pm$ 0.040	0.739 $\pm$ 0.047	0.746 $\pm$ 0.039	0.740 $\pm$ 0.057	0.739 $\pm$ 0.052	0.737 $\pm$ 0.065	0.755 $\pm$ 0.036	0.747 $\pm$ 0.041	0.746 $\pm$ 0.049
Heart	0.876 $\pm$ 0.028	0.703 $\pm$ 0.022	0.711 $\pm$ 0.025	0.739 $\pm$ 0.037	0.728 $\pm$ 0.034	0.789 $\pm$ 0.039	0.859 $\pm$ 0.032	0.864 $\pm$ 0.035	0.851 $\pm$ 0.038
Ionosphere	0.947 $\pm$ 0.027	0.894 $\pm$ 0.024	0.906 $\pm$ 0.021	0.917 $\pm$ 0.017	0.907 $\pm$ 0.025	0.901 $\pm$ 0.027	0.923 $\pm$ 0.029	0.927 $\pm$ 0.020	0.931 $\pm$ 0.026
Libras	0.778 $\pm$ 0.039	0.743 $\pm$ 0.047	0.756 $\pm$ 0.058	0.751 $\pm$ 0.021	0.749 $\pm$ 0.048	0.748 $\pm$ 0.039	0.759 $\pm$ 0.036	0.764 $\pm$ 0.039	0.759 $\pm$ 0.038
Page_small	0.901 $\pm$ 0.019	0.896 $\pm$ 0.015	0.889 $\pm$ 0.013	0.891 $\pm$ 0.016	0.881 $\pm$ 0.019	0.898 $\pm$ 0.017	0.912 $\pm$ 0.015	0.913 $\pm$ 0.011	0.899 $\pm$ 0.012
Parkinsons	0.826 $\pm$ 0.042	0.756 $\pm$ 0.043	0.769 $\pm$ 0.049	0.779 $\pm$ 0.059	0.771 $\pm$ 0.047	0.782 $\pm$ 0.039	0.786 $\pm$ 0.034	0.807 $\pm$ 0.044	0.779 $\pm$ 0.030
Ring	0.928 $\pm$ 0.017	0.781 $\pm$ 0.041	0.831 $\pm$ 0.038	0.849 $\pm$ 0.049	0.834 $\pm$ 0.041	0.879 $\pm$ 0.049	0.870 $\pm$ 0.028	0.910 $\pm$ 0.011	0.925 $\pm$ 0.016
Segment	0.913 $\pm$ 0.016	0.869 $\pm$ 0.019	0.874 $\pm$ 0.014	0.886 $\pm$ 0.018	0.895 $\pm$ 0.016	0.889 $\pm$ 0.020	0.899 $\pm$ 0.015	0.906 $\pm$ 0.013	0.922 $\pm$ 0.010
Sonar	0.816 $\pm$ 0.036	0.758 $\pm$ 0.046	0.769 $\pm$ 0.053	0.779 $\pm$ 0.038	0.806 $\pm$ 0.034	0.787 $\pm$ 0.023	0.796 $\pm$ 0.036	0.801 $\pm$ 0.042	0.837 $\pm$ 0.029
Spectf	0.823 $\pm$ 0.023	0.716 $\pm$ 0.037	0.727 $\pm$ 0.027	0.778 $\pm$ 0.041	0.794 $\pm$ 0.030	0.757 $\pm$ 0.041	0.748 $\pm$ 0.035	0.757 $\pm$ 0.030	0.801 $\pm$ 0.029
Vehicle	0.611 $\pm$ 0.029	0.483 $\pm$ 0.037	0.506 $\pm$ 0.039	0.529 $\pm$ 0.032	0.520 $\pm$ 0.049	0.518 $\pm$ 0.054	0.537 $\pm$ 0.029	0.549 $\pm$ 0.032	0.589 $\pm$ 0.028
Vowel	0.859 $\pm$ 0.023	0.781 $\pm$ 0.057	0.781 $\pm$ 0.052	0.799 $\pm$ 0.058	0.795 $\pm$ 0.057	0.801 $\pm$ 0.033	0.811 $\pm$ 0.029	0.821 $\pm$ 0.024	0.816 $\pm$ 0.027
Wine	0.961 $\pm$ 0.019	0.928 $\pm$ 0.024	0.926 $\pm$ 0.029	0.937 $\pm$ 0.027	0.946 $\pm$ 0.021	0.939 $\pm$ 0.026	0.958 $\pm$ 0.017	0.945 $\pm$ 0.022	0.931 $\pm$ 0.016
WineQR	0.637 $\pm$ 0.018	0.489 $\pm$ 0.025	0.509 $\pm$ 0.038	0.519 $\pm$ 0.037	0.528 $\pm$ 0.034	0.524 $\pm$ 0.031	0.563 $\pm$ 0.028	0.587 $\pm$ 0.030	0.559 $\pm$ 0.041
WineQW	0.534 $\pm$ 0.036	0.477 $\pm$ 0.043	0.503 $\pm$ 0.033	0.508 $\pm$ 0.038	0.501 $\pm$ 0.041	0.509 $\pm$ 0.029	0.508 $\pm$ 0.041	0.515 $\pm$ 0.028	0.529 $\pm$ 0.039
Average	0.800 $\pm$ 0.027	0.730 $\pm$ 0.032	0.740 $\pm$ 0.033	0.756 $\pm$ 0.034	0.755 $\pm$ 0.034	0.756 $\pm$ 0.033	0.768 $\pm$ 0.027	0.777 $\pm$ 0.027	0.781 $\pm$ 0.028

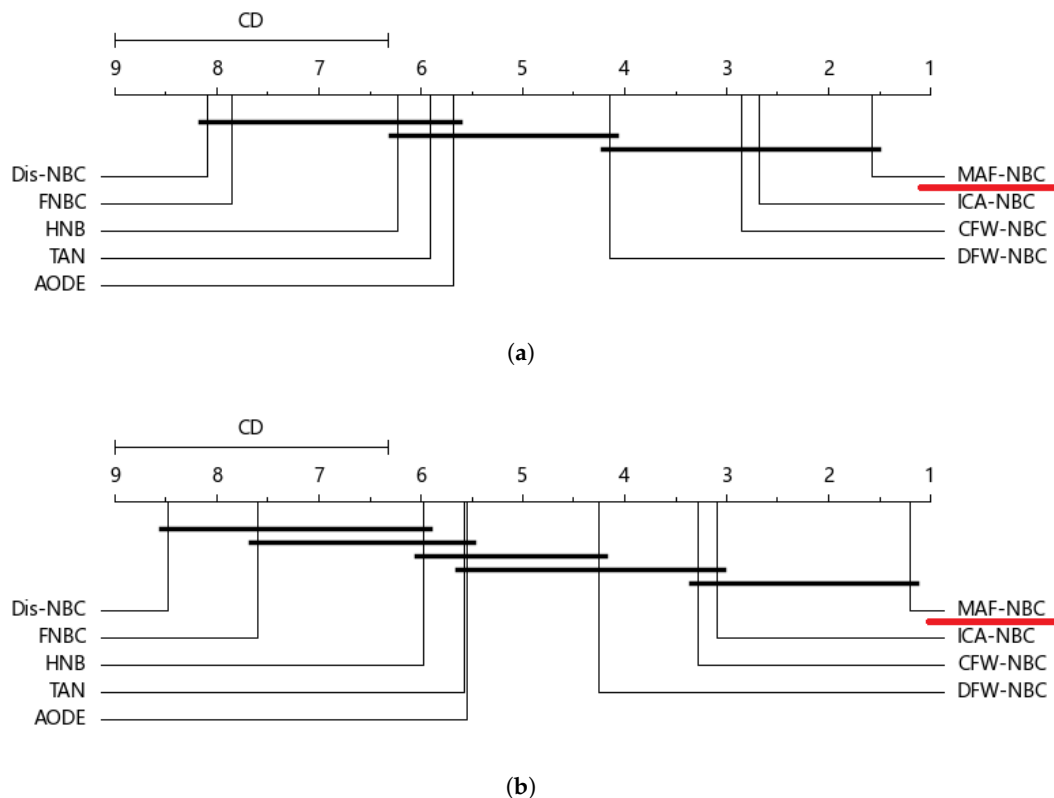


Figure 5. CD diagrams corresponding to comparisons in Tables 2 and 3. (a) CD diagram of training accuracies. (b) CD diagram of testing accuracies.

5. Conclusions and Future Works

This paper presented a novel NBC training method for the mixed-attribute data classification problem without continuous attribute discretization and complex Bayesian network structure learning. The original mixed attributes were transformed into a series of continuous attributes with minimum dependence using an autoencoder neural network. To obtain optimal network weights, an effective objective function was designed, and corresponding weight updating rules were derived. The experimental results finally demonstrated improved classification performance for the novel Bayes model in comparison with eight state-of-the-art Bayesian algorithms. The technical advantages of MAF-NBC are four-fold:

- Preserving the information. The autoencoder neural network transforms mixed attributes into independent encoded attributes, which can also be decoded to restore the original mixed attributes. Thus, useful information from the original data set is conserved as much as possible.
- Simple model structure. MAF-NBC effectively handles the attribute dependence problem between discrete attributes and continuous attributes by transforming original attributes into encoded attributes. The simple structure of the NBC was not modified by providing a data transformation-oriented preprocessing strategy.
- Stable training process. The efficient weight updating strategy guarantees the convergence of the autoencoder neural network and thus provided stable data transformation results. In addition, the quasi-normal distribution of independent encoded attributes provided accurate calculations for the joint conditional probability in the NBC.
- Low computation complexity. A single hidden-layer encoder neural network was used to fuse the original mixed attributes. Training to find optimal network weights has a low computational complexity. As mentioned in Section 3, a shallow learning mechanism was able to satisfy the demand of NBC construction. Hence, it is unne-

essary to explore the availability of deep learning for data transformation-oriented NBC training.

In future studies, we plan (1) to implement MAF-NBC in a distributed environment so that it can be used to deal with large-scale mixed-attribute data-classification problems and (2) to utilize the autoencoder neural network to transform mixed attributes into dependent continuous attributes to construct a non-naive Bayesian classifier [29] based on the joint probability density function estimation technique.

Author Contributions: Data curation, formal analysis, and writing—original draft preparation, G.O.; methodology, writing—original draft preparation and review and editing, Y.H.; investigation and writing—review and editing, P.F.-V.; supervision, J.Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Scientific Research Foundation of National Natural Science Foundation of China (61972261) and Basic Research Foundations of Shenzhen (JCYJ20210324093609026 and JCYJ20200813091134001).

Institutional Review Board Statement: None.

Informed Consent Statement: None.

Data Availability Statement: The data are available in BaiduPan https://pan.baidu.com/s/17417zF-IIP6lW_ut7lRTqA (accessed on 14 October 2022) with extraction code *fcju* or GitHub platform https://github.com/ouguiliang110/DataSet_of_MAF-NBC (accessed on 14 October 2022).

Acknowledgments: The authors would like to thank the editors and anonymous reviewers who carefully read the paper and provided valuable suggestions that considerably improved the paper. This paper was recommended by the 6th Asian Conference on Artificial Intelligence Technology (ACAIT 2022) which is the 2022 Top Academic Conference recognized by China Association for Science and Technology (CAST).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wu, X.; Kumar, V.; Quinlan, J.R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Yu, P.S.; et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [\[CrossRef\]](#)
2. Gil-Begue, S.; Bielza, C.; Larrañaga, P. Multi-dimensional Bayesian network classifiers: A survey. *Artif. Intell. Rev.* **2021**, *54*, 519–559. [\[CrossRef\]](#)
3. John, G.; Langley, P. Estimating continuous distributions in Bayesian classifiers. In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–20 August 1995; pp. 1–8.
4. Friedman, N.; Geiger, D.; Goldszmidt, M. Bayesian network classifiers. *Mach. Learn.* **1997**, *29*, 131–163. [\[CrossRef\]](#)
5. Webb, G.I.; Boughton, J.R.; Wang, Z. Not so naive Bayes: aggregating one-dependence estimators. *Mach. Learn.* **2005**, *58*, 5–24. [\[CrossRef\]](#)
6. Jiang, L.; Zhang, H.; Cai, Z. A novel Bayes model: Hidden naive Bayes. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1361–1371. [\[CrossRef\]](#)
7. Jiang, L.; Li, C.; Wang, S.; Zhang, L. Deep feature weighting for naive Bayes and its application to text classification. *Eng. Appl. Artif. Intell.* **2016**, *52*, 26–39. [\[CrossRef\]](#)
8. Jiang, L.; Zhang, L.; Li, C.; Wu, J. A correlation-based feature weighting filter for Naive Bayes. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 201–213. [\[CrossRef\]](#)
9. Bressan M.; Vitria J. Improving naive Bayes using class-conditional ICA. In Proceedings of the 2022 Ibero-American Conference on Artificial Intelligence, Seville, Spain, 12–15 November 2002; pp. 1–10.
10. Qin, F.; Ren, S.L.; Cheng, Z.K.; et al. Bayes classification model based on ICA. *Comput. Eng. Des.* **2007**, *28*, 4873–4877.
11. Fan, L.; Poh, K.L. A comparative study of PCA, ICA and class-conditional ICA for naive Bayes classifier. In Proceedings of the 2007 International Work-Conference on Artificial Neural Networks, Porto, Portugal, 9–13 September 2007; pp. 16–22.
12. Jayanthi, S.K.; Sasikala, S. Naive Bayesian classifier and PCA for web link spam detection. *Comput. Sci. Telecommun.* **2014**, *41*, 3–15.
13. Zhang, B.; Liu, Z.; Jia, Y.; Ren, J.; Zhao, X. Network intrusion detection method based on PCA and Bayes algorithm. *Secur. Commun. Netw.* **2018**, *2018*, 1914980. [\[CrossRef\]](#)
14. Kurgan, L.A.; Cios, K.J. CAIM discretization algorithm. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 145–153. [\[CrossRef\]](#)

15. Li, M.; Zhao, S.; Chen, S.; Huang, D. A new method of online fault diagnosis based on incremental continuous attribute naive Bayesian. In Proceedings of the 2018 IEEE International Conference on Prognostics and Health Management, Seattle, WA, USA, 11–13 June 2018; pp. 1–7.
16. Scanagatta, M.; Salmerón, A.; Stella, F. A survey on Bayesian network structure learning from data. *Prog. Artif. Intell.* **2019**, *8*, 425–439. [[CrossRef](#)]
17. Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Stat.* **1962**, *33*, 1065–1076. [[CrossRef](#)]
18. Yang, Y.; Webb, G.I. A comparative study of discretization methods for naive-Bayes classifiers. In Proceedings of the 2002 Pacific Rim Knowledge Acquisition Workshop, Tokyo, Japan, 18–22 August 2002; pp. 159–173.
19. Yang, Y.; Webb, G.I. On why discretization works for naive-Bayes classifiers. In Proceedings of the 2003 Australasian Joint Conference on Artificial Intelligence, Perth, Australia, 3–5 December 2003; pp. 440–452.
20. Yang, Y.; Webb, G.I. Discretization for naive-Bayes learning: managing discretization bias and variance. *Mach. Learn.* **2009**, *74*, 39–74. [[CrossRef](#)]
21. He, Y.L.; Ye, X.; Huang, D.F.; Fournier-Viger, P.; Huang, J.Z. A hybrid method to measure distribution consistency of mixed attribute data sets. *IEEE Trans. Artif. Intell.* **2022**. [[CrossRef](#)]
22. Qi, Z.; Wang, B.; Tian, Y.; Zhang, P. When ensemble learning meets deep learning: A new deep support vector machine for classification. *Knowl.-Based Syst.* **2016**, *107*, 54–60. [[CrossRef](#)]
23. Feng, J.; Yu, Y.; Zhou, Z.H. Multi-layered gradient boosting decision trees. In Proceedings of the 32nd Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 1–11.
24. Guo, M.; Fei, J.; Meng, Y. Deep nearest neighbor website fingerprinting attack technology. *Secur. Commun. Netw.* **2021**, *2021*, 5399816. [[CrossRef](#)]
25. De Freitas, J.F.; Niranjan, M.; Gee, A.H.; Doucet, A. Sequential Monte Carlo methods to train neural network models. *Neural Comput.* **2000**, *12*, 955–993. [[CrossRef](#)] [[PubMed](#)]
26. Alcalá-Fdez, J.; Fernández, A.; Luengo, J.; Derrac, J.; García, S.; Sánchez, L.; Herrera, F. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J.-Mult.-Valued Log. Soft Comput.* **2011**, *17*, 255–287.
27. Helal, A.; Otero, F.E. A mixed attribute approach in ant-miner classification rule discovery algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference, Denver, CO, USA, 20–24 July 2016; pp. 13–20.
28. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
29. Wang, X.Z.; He, Y.L.; Wang, D.D. Non-naive Bayesian classifiers for classification problems with continuous attributes. *IEEE Trans. Cybern.* **2013**, *44*, 21–39. [[CrossRef](#)] [[PubMed](#)]