UNIVERSITY OF NOTTINGHAM NINGBO CHINA

FACULTY OF SCIENCE AND ENGINEERING

Department of Computer Science

Enhanced Naïve Bayes Classification Framework with Data Reduction and Transformation Techniques

Wang Shihe

Supervised by

Dr. Ren Jianfeng

Prof. Bai Ruibin

Dr. Yao Yuan

Prof. Liu Tieyan

China, 2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Name: ....

Date: ....

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisor, Associate Professor REN Jianfeng, for his guidance, patience and encouragement. His inspiriting nature and continuous encouragement have been a great source of motivation throughout my doctoral study. His sharp insights, excellent guidance and strict attitude toward research have been a model for me and will continuously benefit me throughout my lifetime.

I also would like to take this opportunity to express my sincere gratitude to my co-advisor Professor BAI Ruibin. His insightful advice and warmly encouragement greatly help my research.

I sincerely thank my friends, Ms. Danzhao Cheng, Ms. DING Ruxin, Ms. GU Xinyu, Mr. HE Wentao, Mr. SONG Xingke, Mr. YAO Chenglin, Ms. ZHANG Jialu and Mr. ZHANG Zibo, for their encouragements and helps.

I sincerely thank the School of Computer Science, University of Nottingham Ningbo China, for offering me the opportunity of doctoral study. I also sincerely thank my co-advisor, Professor Liu Tieyan, together with Digital Port Technologies Lab and Microsoft Research Asia for offering me the research-associate position so that I have financial support to complete my doctoral study.

Finally, I would like to dedicate this work to my parents and my brother for their persistent love, continuous encouragement and endless support.

Abstract

The Bayesian classification framework has been widely used in many fields, but the covariance matrix is usually difficult to estimate reliably. To alleviate the problem, many naive Bayes (NB) approaches with good performance have been developed. However, the assumption of conditional independence between attributes in NB rarely holds in reality. Various attribute-weighting schemes have been developed to address this problem. Among them, class-specific attribute weighted naive Bayes (CAWNB) has recently achieved good performance by using classification feedback to optimize the attribute weights of each class. However, the derived model may be over-fitted to the training dataset, especially when the dataset is insufficient to train a model with good generalization performance. Moreover, the Bayesian classification framework often relies on the discretization method to handle the various data types. Existing discretization methods often target maximizing the discriminant power of discretized data, while overlooking the fact that the primary target of data discretization in classification is to improve the generalization performance. As a result, the data tend to be over-split into many small bins since the data without discretization retain the maximal discriminant information.

In this thesis, we exploit the data intrinsic by using data reduction and transformation methods. In Chapter 3, we propose a regularization technique to improve the generalization capability of naive Bayes classifier, which could well balance the trade-off between discrimination power and generalization capability. We boost the discriminant power of naive Bayes by developing a semi-supervised discretization framework with an adaptive discriminative selection criterion in Chapter 4. Besides, a well-designed discretization scheme using a Max-Relevancy-Min-Divergence (MRmD) criterion is introduced to better balance the generalization ability and discrimination power of subsequent classifiers discussed in Chapter 5. To reduce the data noise and alleviate the weakness in capturing the feature correlation, a feature augmentation framework employing the stacked autoencoder is proposed in Chapter 6. These contributions are discussed in detail as follows.

Firstly, we propose a regularization technique to improve the generalization capability of naive Bayes classifier, which could well balance the trade-off between discrimination power and generalization capability. More specifically, by introducing the regularization term, the proposed method, namely regularized naive Bayes (RNB), could well capture the data characteristics when the dataset is large, and exhibit good generalization performance when the dataset is small. RNB is compared with the state-of-the-art naive Bayes methods. Experiments on 33 machine-learning benchmark datasets demonstrate that RNB outperforms other NB methods significantly.

Secondly, we design a semi-supervised adaptive discriminative discretization (SADD) scheme to address the significant information loss in previous discretization methods and improve the performance of naive Bayes classifiers. To make full use of labeled and unlabeled data, the pseudo-labeling technique is utilized to compute the pseudo labels for unlabeled data. Then, an adaptive discriminative selection criterion is proposed to further reduce the information loss and the resulting discretization scheme could achieve a better trade-off between generalization ability and discrimination power. Experimental results on 31 machine-learning datasets validate the effectiveness of the proposed SADD.

Thirdly, we propose a Max-Dependency-Min-Divergence (MDmD) criterion that maximizes both the discriminant information and generalization ability of the discretized data, and hence the performance of NB classifier can be improved. More specifically, the Max-Dependency criterion maximizes the statistical dependency between the discretized data and the classification variable while the Min-Divergence criterion explicitly minimizes the JS-divergence between the training data and the validation data for a given discretization scheme. The proposed MRmD is compared with the state-of-the-art discretization algorithms under the naive Bayes classification framework on 45 machine-learning benchmark datasets. It significantly outperforms all the compared methods on most of the datasets.

Fourthly, we enhance the discriminant power of NB classifiers by a stack auto-encoder that consists of two auto-encoders for different purposes. The first encoder shrinks the initial features to derive a compact feature representation in order to remove the noise and redundant information. The second encoder boosts the discriminant power of the features by expanding them into a higher-dimensional space so that different classes of samples could be better separated in the higher-dimensional space. By integrating with the state-of-the-art NB method, regularized naive Bayes (FAR-NB), the discrimination power of the model is greatly enhanced. The proposed FAR-NB is compared with the state-of-the-art NB classifiers and achieves a superior classification performance.

The contributions of this thesis are summarized as follows:

- We propose a regularized naive Bayes classifier to automatically balance the generalization ability and discrimination power by optimizing the attribute weights.
- We propose a semi-supervised adaptive discriminative discretization scheme to reduce the significant information loss in state-of-the-art naive Bayes classifiers.
- We propose to boost the performance of NB classifier from a discretization perspective, using a Max-Relevancy-Min-Divergence discretization scheme.
- We propose a feature augmentation method to enhance the discrimination power of NB classifier employing stack autoencoder to explore the data intrinsic residing in the original space.

Contents

List of Figures

List of Tables

List of Acronyms

AE Autoencoder.

AIWNB Attribute and Instance Weighted Naive Bayes.

AODE Averaged One-Dependence Estimators.

ATAN Averaged Tree Augmented Naive Bayes.

AUC Area Under Curve.

CACC Class-Attribute Contingency Coefficients.

CAIM Class-Attribute Interdependence Maximization.

CAWNB Class-specific Attribute Weighted Naive Bayes.

CFW Correlation-based Feature weiggting.

CLL Conditional Log-Likelihood.

CNNs Convolutional Neural Networks.

DAWNB Deep Feature Weighting for Naive Bayes.

DEAWNB Differential Evolution for Attribute Weighted Naive Bayes.

DWNB Discriminatively Weighted Naive Bayes.

EMD Evolutionary Multivariate Discretization.

FAR-NB Feature Augmentation for Regularized Naive Bayes.

FFD Fixed Frequency Discretization.

GNNs Generative Adversarial Networks.

k-NN k-Nearest Neighbors.

KL Kullback-Leibler.

LWNB Locally Weighted Naive Bayes.

MDLP Minimum Description Length Principle.

MDmD Max-Dependency-Min-Divergence.

MDS Multi-Dimensional Scaling.

MRmD Max-Relevance-Min-Divergence.

MSE Mean Square Error.

NB Naive Bayes.

PCA Principal Component Analysis.

PKID Proportional K-Interval Discretization.

RNB Regularized Naive Bayes.

SADD Semi-supervised Adaptive Discriminative Discretization.

SMOTE Synthetic Minority Over-sampling Technique.

SODE Self-adaptive One-dependence Estimators.

t-SNE t-distributed Stochastic Neighbor Embedding.

TAN Tree Augmented Naive Bayes.

TCSFS-NB Test-Cost-Sensitive Feature Selection for Naive Bayes.

WANBIA Weighting attributes to Alleviate Naive Bayes' Independence Assumption.

Chapter 1

Introduction

This thesis focuses on improving the naive Bayes classification framework through feature weighting, data discretization and data augmentation methods. The motivations of our research are explained in Section 1.1. The major contributions of this thesis are summarized in Section 1.2. Finally, the organization of this work is described in Section 1.3.

1.1 Motivation

The Bayesian rule is widely used in classification, clustering and regression for building probabilistic models. The Bayesian approach provides a flexible and theoretical way to measure the uncertainty in various applications including natural language processing [1], computer vision [2] and bioinformatics [3]. In classification tasks, Bayesian classification provides a probabilistic framework that explicitly represents uncertainty and facilitates the integration of prior knowledge [4]. This characteristic enhances the capacity for making well-informed and robust classification decisions, particularly in contexts where training data is scarce. Furthermore, Bayesian classifiers exhibit capability in addressing scenarios characterized by imbalanced class distributions or missing data due to their probabilistic formulation. Despite its advantages, the Bayesian learning approach can be computationally expensive while dealing with high-dimensional data. Besides, it's difficult to reliably estimate the joint probability due to the curse of dimensionality.

Hence, naive Bayes has been developed to alleviate this problem by assuming that the features are independent of each other. Naive Bayes is a simple and efficient machine-learning algorithm widely applied in text classification [5], malware detection [6], and recommendation systems [7]. Besides, naive Bayes could handle both continuous and categorical features. For continuous features, naive Bayes often assumes that the values of each feature given class variable follow the Gaussian distribution. Hence, the mean and variance are utilized to estimate the likelihood probability of the data. For categorical features, the likelihood probability of each category given the class variable is estimated by calculating the frequency of the data.

Compared with other traditional learning algorithms and deep learning-based methods, naive Bayes could achieve competitively higher performance with real-time efficiency [8]. Besides, naive Bayes has high explainability, few parameters and robustness to noisy or missing data [9]. Due to high scalability, it can well handle datasets with various sizes including large datasets with millions of samples and thousands of features and small datasets with few samples and features [10]. However, the independence assumption is often violated in real-world applications in cases where there exist correlations between features. To address this problem, various improvements on naive Bayes have been developed, which can be broadly divided into five categories: structure extension [11, 12], instance selection [13], instance weighting [14, 15], Feature selection [16–18] and feature weighting [9, 19, 20]. Among them, feature weighting methods achieve superior performance by assigning different weights to different features so that the discriminative feature will have a larger weight. However, they either define class-independent weights to emphasize the generalization ability [21] or class-dependent weights to emphasize the discrimination power [22]. In classification tasks, both generalization ability and discrimination power are often jointly considered to derive the optimal learning model. In Chapter 3, we proposed a regularized attribute weighting framework by constraining the class-dependent weights with the class-independent ones to automatically balance the generalization ability and discrimination power.

Another problem of naive Bayes is that the probability distribution of continuous features is assumed to follow the Gaussian distribution. However, the data doesn't always fit Gaussian distribution which may be uniform distribution, Poisson distribution or any others. To address this problem, the continuous data is often dicretized into discrete one and hence naive Bayes can handle it similarly to the categorical data. For example, MDLP discretizer selects the cut points by maximizing the entropy of the data and designs a stop criterion to prevent over splitting. MDLP has been widely used in advanced naive Bayes classifiers and yields satisfactory performance [9, 20–22]. However, MDLP often discretizes data into a small number of intervals that result in significant information loss. Existing discretization methods often target maximizing the discriminant power of discretized data, while overlooking the fact that the primary target of data discretization in classification is to improve the generalization performance. As a result, the data tend to be over-split into many small bins since the data without discretization retain the maximal discriminant information [23–25]. Thus we proposed a semi-supervised discretization framework to better preserve the discriminant information, and proposed a maximial-relevancy-minimal-divergence discretization criterion to simultaneously maximize the discriminant information and the generalization ability of discretized data. The two proposed methods are discussed in Chapter 4 and Chapter 5 in detail, respectively.

As naive Bayes handles each feature dimension separately, it lacks a mechanism to model the correlations between features. Besides, the local data structure formed by jointly considering all the feature dimensions of neighboring samples is often disrupted when each feature dimension is handled separately in naive Bayes. Due to these challenges, the discriminant power of naive Bayes is often undermined. Although many state-of-the-art naive Bayes alleviate the impact of feature correlation via structure extension, feature weighting and feature selection. It is a lack of a mechanism to explicitly capture the correlation information between features. To capture the correlation information between features, there are many dimensionality reduction methods including Principal Component Analysis (PCA) [26], t-distributed Stochastic Neighbor Embedding (T-SNE) [27], Multi-Dimensional Scaling (MDS) [28] and autoencoder (AE) [29]. However, reducing the feature dimensions will lead to information loss and hence degrade the discrimination power of subsequent classification models. Thus, a feature augmentation framework is proposed to enhance the discrimination power of naive Bayes classifiers. This work is discussed in Chapter 6 in detail.

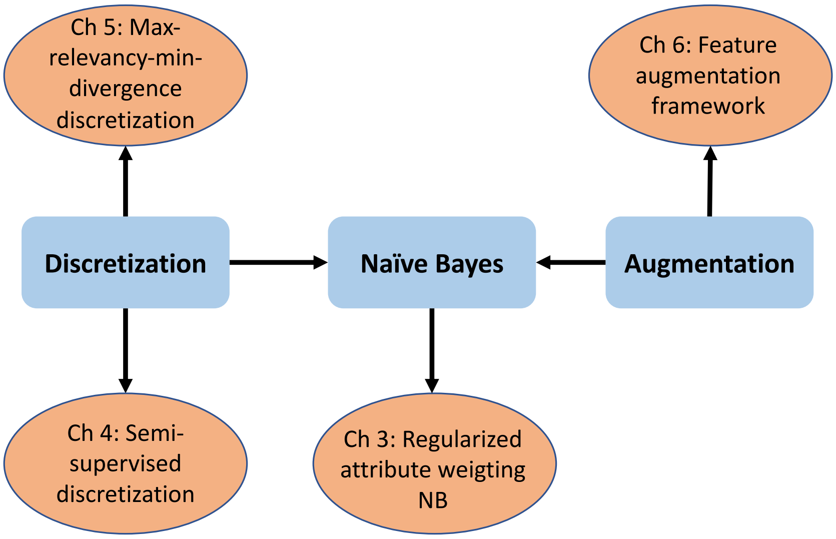

Ch 5: Max-relevancy-min-divergence discretization
Ch 6: Feature augmentation framework
Discretization
Naïve Bayes
Augmentation
Ch 4: Semi-supervised discretization
Ch 3: Regularized attribute weigting NB

Figure 1.1: The block diagram of the thesis.

1.2 Contributions

The major contributions of this thesis are summarized in Fig. 1.1. They can be grouped into three aspects: 1). Improvement of attribute weighting framework on naive Bayes; 2). Improvement of naive Bayes methods on discretization 3). improvement of naive Bayes on augmentation. These contributions are listed as follows.

- To improve the generalization capability of CAWNB, we propose to add a simpler model, WANBIA, to constrain CAWNB. Besides, CAWNB is an improved version of WANBIA, and both share a similar optimization procedure. It will not significantly increase the computational complexity by integrating WANBIA into CAWNB. Thus, a regularized attribute-weighting framework is proposed to automatically balance the generalization ability and discrimination power of the naive Bayes classification model (RNB).
- A Max-Dependency-Min-Divergence criterion (MDmD) is proposed to simultaneously maximize the discriminant power and minimize the distribution discrepancy so that the derived discretization scheme could generalize well to the data population. To tackle the challenges of reliable estimation of the joint probabilities in MDmD, a more practical solution, the Max-Relevance-Min-Divergence (MRmD)

discretization scheme, is proposed to derive the optimal discretization scheme for one attribute at a time. Sequentially, the naive Bayes can be improved through a better trade-off between generalization ability and discrimination power in discretization.

- A semi-supervised adaptive discriminative discretization (SADD) is proposed to address a significant information loss of previous state-of-the-art naive Bayes methods. The proposed SADD could better estimate the data distribution by utilizing both labeled data and unlabeled data through pseudo-labeling techniques and significantly reduces the information loss during discretization with adaptive discriminative discretization scheme, and hence greatly improves the discrimination power of NB classifiers.
- To reduce the data noise and capture the feature correlation, we proposed a feature augmentation framework for regularized naive Bayes (FAR-NB) method. FAR-NB utilizes the stacked autoencoder to capture the correlation into the learned feature representations, in which the original feature is firstly shrunk into a compact representation and then the compact code is expanded into the higher-dimensional space to boost the discriminant power. Finally, the input feature, the learned feature and the reconstructed feature are concatenated into a feature set to build the regularized naive Bayes classification model.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows.

- In Chapter 2, the representative naive Bayes classifiers, data discretization methods and feature augmentation methods are reviewed.
- In Chapter 3, a regularized attribute weighting framework for naive Bayes is proposed for automatically balancing the generalization ability and discrimination power.

- In Chapter 4, a semi-supervised adaptive discriminative discretization is proposed to reduce the significant information of previous state-of-the-art naive Bayes classifiers.
- In Chapter 5, a Max-Relevancy-Min-Divergence (MRmD) criterion that maximizes both the discriminant information and generalization ability of the discretized data is proposed and applied to the naive Bayes classification framework.
- In Chapter 6, a feature augmentation method is proposed to enrich the discriminant information of the data, and has improved the performance of the naive Bayes classifier.
- In Chapter 7, we conclude this thesis by highlighting our contributions and discussing possible future work.

Chapter 2

Literature Review

2.1 Bayesian Classification Framework

Bayesian classification framework is a probabilistic machine learning model based on Bayes' Theorem [30]. Bayesian classification addresses the classification problem by learning the distribution of instances given different class variables which estimate the joint probability distribution of the class variable and the attributes [31]. It states that the posterior probability of a class given the data is proportional to the likelihood of the data given the class and the prior probability of the class. The posterior probability is defined as:

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c)P(c)}{P(\boldsymbol{x})}, \tag{2.1}$$

where $\boldsymbol{x}$ is the feature vector and c is the classification variable, $P(\boldsymbol{x}|c)$ is the likelihood probability, $P(c)$ is prior probability, $P(\boldsymbol{x})$ is the marginal probability. Because it is difficult to reliably estimate the likelihood $P(\boldsymbol{x}|c)$ due to the curse of dimensionality, in naive Bayes methods, the likelihood is estimated by assuming that the attributes are independent given the classification variable c, which results in the following formulation:

$$P(\boldsymbol{x}|c) = \prod_{j=1}^{m} P(x_j|c), \tag{2.2}$$

where x_j is the j-th dimension of the feature vector $\boldsymbol{x}$, and m is the feature dimensionality. Then, the posterior probability can be estimated by:

$$P(c|\boldsymbol{x}) = \frac{P(c) \prod_{j=1}^{m} P(x_j|c)}{\sum_{c'} P(c') \prod_{j=1}^{m} P(x_j|c')}. \tag{2.3}$$

Once the probabilistic model is derived, the prior probability and the likelihood probabilities are used to estimate the posterior probability of the novel data $\boldsymbol{t}$. Finally, the maximum a posterior (MAP) Estimation is often used to make the classification:

$$\hat{c}(\boldsymbol{t}) = \underset{c \in \boldsymbol{C}}{\arg\max}\, \hat{P}(c|\boldsymbol{t}), \tag{2.4}$$

where $\boldsymbol{C}$ is the set of labels for all classes. The class variable c that gives the highest classification.

By ignoring the inter-feature dependencies, it may lead to the loss of discriminant information and poor estimation of likelihood and hence result in low discrimination power. Many advanced naive Bayes have been developed to alleviate this problem including structure extension [11, 12], selection [13, 15, 17, 18] and weighting [5, 19, 21, 22, 32–43]. Among them, we mainly investigate the feature weighting methods and identify the weakness of emphasizing on discrimination power while overlooking the generalization ability [5, 19, 21, 22, 32–42]. Hence, we have explored to balance the discrimination power and generalization of naive Bayes classification framework through data reduction and transformation such as feature weighting [21, 22], discretization [44, 45] and augmentation [46–48]. In the following sections, we will mainly review the existing state-of-the-art naive Bayes methods, data discretization methods and augmentation methods.

2.2 Naive Bayes

Naive Bayes (NB) classifiers have been extensively utilized in a wide range of applications [49–51]. However, the strong assumption of feature independence in NB is frequently violated in real-world datasets. To address this problem, numerous enhancements have been proposed, which can be broadly classified into five categories. The first category, structure extension [11, 12], modifies the structure of the naive Bayes model to represent dependencies among features, thereby improving its capacity to capture relationships between features. The second category, instance selection [13, 15], employs the principle of local learning to construct a set of local naive Bayes classifiers using subsets of the dataset. This approach enhances the model's adaptability to local data variations. The third category, instance weighting [43], assigns different weights to instances to maximize the discriminant power of the classifier. Feature selection [17, 18], the fourth category, focuses on eliminating strongly correlated or irrelevant features that could undermine the reliability of the classification. By selecting the most discriminative subset of features, this approach enhances the performance and robustness of classification model. The fifth category, attribute weighted naive Bayes [5, 19–22, 32–42], tackles the independence assumption by assigning different weights to attributes. This method increases the weight of discriminative features, thereby enhancing the overall discriminative power of the model. Table 2.1 presents an overview of state-of-the-art naive Bayes methods, detailing the advantages and limitations of each category.

Table 2.1: Overview of state-of-the-art naive Bayes methods.

Category	**Method**	**Advantage**	**Limitation**
Structure Extension	Tree-Augmented Naïve Bayes (TAN) [52]	- Models feature dependencies to enhance discrimination power	- Increased computational complexity - Low scalability on large dataset
	Averaged One-Dependence Estimators (AODE) [53]		
	Averaged Tree Augmented Naïve Bayes [54]		
	Hidden Naïve Bayes (HNB) [55]		
	Self-adaptive One-dependence Estimator [11]		
Instance Selection	Naïve Bayes Tree [56]	- Reduces noise in training data	- Loss of discriminative information - Increased risk of overfitting
	Multi-variate Bernoulli Naïve Bayes (BNB) [13]		
	Multinominal Naïve Bayes (MNB) [13]		
	Locally Weighted Naive Bayes [57]		
Instance Weighting	Discriminatively Weighted Naïve Bayes (DWNB) [43]	- Enhance the discriminative ability	- High computational complexity
	Attribute Value Frequency-based Instance Weighted Naive Bayes (AVFWNB) [14]		
Feature Selection	Randomly Selected Naïve Bayes [58]	- Balance model simplicity and accuracy	- Potential loss of feature interactions
	Selective Bayesian classifier (SBC) [59]		
	Selective Naive Bayes [60]		
Feature Weighting	Decision Tree-based Feature Weighting (DTFW) [33]	- Enhance discrimination power - Reduce the effect of irrelevant features	- Loss of generalization ability - Computational expensive
	Deep Feature Weighting (DFW) [32]		
	Attribute and Instance Weighted Naïve Bayes (AIWNB) [20]		
	Weighting to Alleviate Naïve Bayes Independence Assumption (WANBIA) [21]		
	Class-specific Attribute Weighted Naïve Bayes (CAWNB) [22]		

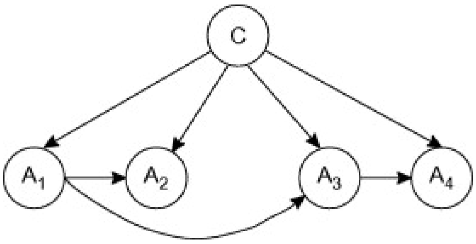


FIGURE 2.1: Example structure of TAN.

2.2.1 Structure-extension Naive Bayes

Among the improved naive Bayes methods, structure extension is the most direct way to improve Naive Bayes, since attribute dependencies can be explicitly represented by arcs. Tree Augmented Naive Bayes (TAN) is an extended tree-like Naive Bayes [52]. Unlike the standard naive Bayes, in TAN, the class node directly points to all attribute nodes and each attribute node has at most one parent from another attribute node. TAN is a specific case of general Bayesian network classifiers [61], in which the class node directly points to all attribute nodes and each attribute node can point to the other ones. In practice, TAN is a good trade-off between model complexity and learnability. To build the TNB structure, the conditional mutual information $I(A_i; A_j|C), i \neq j$ between each pair of attributes is computed. Then, a complete undirected graph is built in which nodes are attributes $A_1, \ldots, A_m$ and the weight of an edge connecting A_i to A_j is estimated by $I(A_i; A_j|C)$. The derived undirected tree is then transformed into a directed one by choosing a root attribute and setting the direction of all edges to be outward from it. Finally, the full TAN structure is derived by adding the node of the class variable C and adding the arcs pointing to all attribute nodes. Fig. 2.1 shows an example of TAN.

In [62], a greedy heuristic search algorithm is developed to improve the classification accuracy of TAN by finding the optimal arcs between attributes. Zhang and Ling [63] observed that attributes tend to be dependent and cluster into groups. Based on this finding, an efficient searching algorithm has been developed to speed up the graph-building process while maintaining similar classification accuracy. In [53], an averaged one-dependence estimator (AODE) is developed by constructing the TAN for each attribute, in which the attribute is directly set to be the parent of all the other attributes.

Then, AODE produces the prediction by directly averaging the predictions of all qualified TAN classifiers. Thus, AODE can be viewed as an ensemble learning algorithm. Similar in [54], an averaged tree augmented naive Bayes (ATAN) is developed to build multiple TAN classifiers by regarding each attribute as the root node and hence improve the generalization ability of a single TAN classifier. For the ATAN classifier, the posterior probability is defined as:

$$P_A(c|\boldsymbol{x}) = \frac{1}{m}\sum_{i=0}^{m} P(c|\boldsymbol{x})_i, \tag{2.5}$$

where m is the number of attributes. To improve the discrimination power of the classifier, a weighted ATNB is developed:

$$P_A(c|\boldsymbol{x}) = \frac{1}{\sum w_i}\frac{1}{m}\sum_{i=0}^{m} w_i P(c|\boldsymbol{x})_i, \tag{2.6}$$

where $\boldsymbol{w}$ is a weight vector estimated by using the mutual information between A_i and class C:

$$I(A_i|C) = \sum_{a\in A_i}\sum_{c\in C} P(a,c)\frac{P(a,c)}{P(a)P(c)}. \tag{2.7}$$

Besides the ensemble methods, the hidden naive Bayes (HNB) relieves the structure of TAN by adding the hidden parent for each attribute [55]. The posterior of HNB is defined as:

$$P(A_i|A_{hp_i}, C) = \sum_{j=1, j\neq i}^{n} W_{ij} * P(A_i|A_j, C), \tag{2.8}$$

where A_{hpi} is the hidden parent node for A_i, which is a mixture of the weighted influences from all other attributes by using a weight matrix $\boldsymbol{W}$. W_{ij} is defined as:

$$W_{ij} = \frac{I(A_i; A_j|C)}{\sum_{j=1, j\neq i}^{n} I(A_i; A_j|C)}. \tag{2.9}$$

Recently, a self-adaptive one-dependence estimator (SODE) is developed based on the one-dependence estimator (ODE) method to dynamically optimize the weights for multiple ODEs [11]. In a word, extending the structure of NB can mitigate the conditional independence assumption to some extent [11], but it is a rather difficult problem to obtain a suitable structure of extended NB. Also, the structure extension method is computationally intensive.

2.2.2 Instance-selection Naive Bayes

The main idea of the instance selection method is to build a Naive Bayes model on a subset of the training instances instead of using the entire dataset. While the assumption of conditional independence may not hold true for the entire training dataset, creating a local NB model on a smaller dataset that is in the proximity of the test instance may yield better results. Instance selection can also be further divided into eager learning [13, 56] and lazy learning [20]. Eager learning learns a model from the training data before making predictions on new data. Otherwise, lazy learning delays the processing of training data until the time of prediction.

Instance selection with eager learning includes NB tree [56] and multinomial naive Bayes tree [13]. In [56], a hybrid model of naive Bayes classifier and decision tree called NBTree is developed in which the NBTree is first constructed to split attributes similar to C4.5 by using 5-fold cross-validation [56]. At each leaf, a local naive Bayes is built to make a prediction for a new instance. In [13], an adaptive naive Bayes tree is presented by assigning different weights to different attributes to improve the text classification performance. In text classification, the multi-variate Bernoulli naive Bayes (BNB) model is often used. Given a documentation d represented by a binary word vector $w_i, \ldots, w_m$, the conditional probability $P(d|c)$ is estimated by:

$$P(d|c) = \prod_{i=1}^{m} \left(w_i P(w_i|c) + (1 - w_i)(1 - P(w_i|c)) \right), \tag{2.10}$$

where m is number of words, w_i is a boolean value which represents whether the ith word appears in d or not, and the conditional probability $P(w_i|c)$ is estimated by,

$$P(w_i|c) = \frac{\sum_{j=1}^{n} w_{ji}\delta(c_j, c) + 1}{\sum_{j=1}^{n} \delta(c_j, c) + 2}. \tag{2.11}$$

where n is the number of training documents, c_j is the class label of the j-th training document, w_{ji} is the i-th word of the jth training document, and $\delta(\bullet)$ is a binary function, which is one if its two parameters are identical and zero otherwise. To improve the discrimination power of BNB, the multinominal naive Bayes (MNB) model [13] captures the frequencies that all of the words occur in a document in which the conditional probability is estimated by,

$$P(d|c) = \left(\sum_{i=1}^{m} f_i\right)! \prod_{i=1}^{m} \frac{P(w_i|c)^{f_i}}{f_i!}, \tag{2.12}$$

where f_i is the frequency count of w_i in the document d, $P(w_i|c)$ is the conditional probability that the word w_i occurs in the class c, which can be estimated by,

$$P(w_i|c) = \frac{\sum_{j=1}^{n} f_{ji} w_{ji} \delta(c_j, c) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ji} \delta(c_j, c) + m}. \tag{2.13}$$

where f_{ji} is the frequency count of the word w_i in the j-th training document. Instead of using classification accuracy to build the tree, MNB utilizes the information gain to select the optimal attribute as the split attribute. Given the documentation set D, the information gain using the word w_i to partition D can be defined as,

$$Gain(D, w_i) = Entropy(D) - \sum_{v \in 0, \bar{0}} \frac{|D_v|}{|D|} Entropy(D_v), \tag{2.14}$$

where $|D_v|$ is the number of the instances whose value of the attribute w_i is v ($v \in 0, \bar{0}$), $Entropy(D)$ is the entropy of D, which can be calculated by,

$$Entropy(D) = -\sum_{c \in C} P(c) log P(c). \tag{2.15}$$

Except for eager learning, lazy learning is another way to select the instances to build a local naive Bayes classifier. The k-nearest neighbor (k-NN) algorithm is one of the simple and efficient methods for local learning. In [57], a locally weighted naive Bayes (LWNB) is introduced, which constructs a local NB according to the k-nearest neighbors of the test instance. LWNB enhances NB in classification accuracy, and it is not highly sensitive to the value of k unless k is excessively small.

2.2.3 Instance-weighting Naive Bayes

Instance weighting is an improved version of instance selection by assigning different weights to different instances. To estimate the posterior probability, the prior probability and the likelihood probability are defined as:

$$P(c) = \frac{\sum_{j=1}^{n} w_i^{ins}\delta(c_j, c) + \frac{1}{q}}{\sum_{j=1}^{n} w_i^{ins} + 1}, \tag{2.16}$$

$$P(a_j|c) = \frac{\sum_{i=1}^{n} w_i^{ins}\delta(a_{ij}, a_j)\delta(c_i, c) + \frac{1}{n_j}}{\sum_{i=1}^{n} w_i^{ins}\delta(c_i, c) + 1}, \tag{2.17}$$

where w_i^{ins} is the weight of the ith training instance, q is the number of classes and n_j is number of attribute values for jth attribute A_j. Similarly, the instance weighting method also can be divided into eager learning methods and lazy learning methods [20]. In [43], Jiang *et al.* presented a discriminatively weighted naive Bayes (DWNB) in an eager learning way by estimating the instance weight for ith instance as:

$$\boldsymbol{w}_i^{ins} = 1 - \hat{P}(c|\boldsymbol{x}_i). \tag{2.18}$$

where $\hat{P}(c|\boldsymbol{x}_i)$ is the conditional probability of instance $\boldsymbol{x}_i$ given the class variable c based on whole training set. Then, the instance weights are updated for a few iterations until converge. To improve the efficiency of the learning process, Xu *et al.* developed an attribute value frequency-based instance weighted naive Bayes (AVFWNB) based on the frequency of the attribute value rather than the attribute value itself [14]. At first, the frequency of each attribute value is estimated by:

$$f_{ij} = \frac{\sum_{k=1}^{n} \delta(a_{kj}, a_{ij})}{n}, \tag{2.19}$$

where f_{ij} is the frequency of a_{ij} which is the jth attribute value of the ith training instance, n is the number of training instances, a_{kj} is the jth attribute value of the kth training instance. Then, let n_j be the number of attribute values in jth attribute, the weight of ith training instance is estimated by:

$$\boldsymbol{w}_i = \sum_{j=1}^{m} f_{ij} * n_j \tag{2.20}$$

In [64], a lazy naive Bayes model is introduced by weighting each training instance according to the similarity between the test instance $\boldsymbol{t}$ and each training instance $\boldsymbol{x}$:

$$s(\boldsymbol{x}, \boldsymbol{t}) = \sum_{i=1}^{n} \delta(x_i, t_i). \tag{2.21}$$

where $s(\boldsymbol{x}, \boldsymbol{t})$ simply counts the number of identical attribute values between $\boldsymbol{x}$ and $\boldsymbol{t}$. Then, the training instance is duplicated a number of times according to its similarity to the test instance. Finally, a NB classifier is built using the expanded training set.

2.2.4 Feature-selection Naive Bayes

Feature selection in naive Bayes is a process of selecting a subset of features from the original set of features. By identifying and removing the irrelevant and redundant features, the efficiency and accuracy of the naive Bayes classifier can be improved. Different from standard naive Bayes, the feature selection-based naive Bayes methods make a prediction on the subsets of features and hence the Eqn. 3.10 is adapted to:

$$\hat{c}(\boldsymbol{t}) = \arg\max_{c \in \boldsymbol{C}} \prod_{j=1}^{s} \hat{P}(c|t_j). \tag{2.22}$$

where s is the number of features in the selected subset. To find the optimal feature subset, many attribute selection-based method is developed, which can be further divided into filter-based methods [59] and wrapper-based methods [58]. Filter-based methods utilize a set of evaluation criteria to determine the feature subset, while wrapper-based methods utilize classification feedback to optimize the selection process. Wrapper methods often provide better classification performance than filter ones, but at a higher computational cost. Both filter-based and wrapper-based methods follow similar selection processes as shown in Fig. 2.2. In this process, the evaluation criterion and search strategy are the two most important parts on which many feature selection methods focus [58].

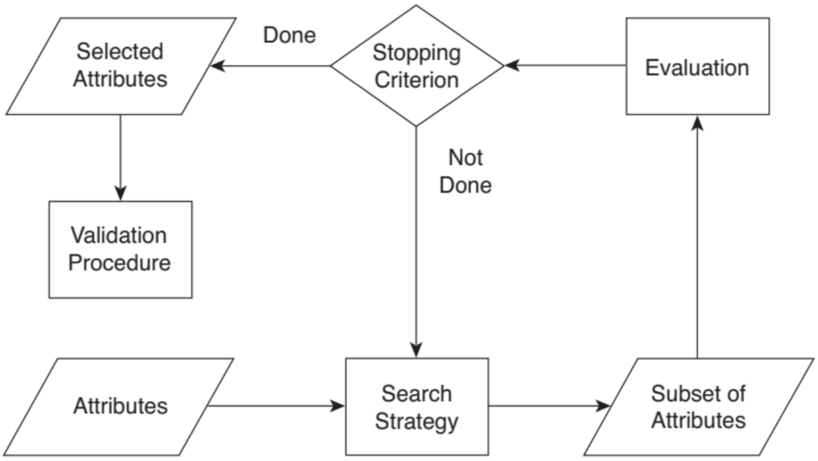


FIGURE 2.2: The block diagram of feature selection.

2.2.4.1 Filter-based Methods

In filter-based methods, many feature evaluation functions have been developed including consistency and correlation, information gain, mutual information and distance metrics [65–68]. Pearson's correlation is one of the commonly used statistical evaluation criteria. [68]. To measure the linear relationship between two random variables X and Y, the Pearson correlation is defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{2.23}$$

where n is the number of samples in X and $\bar{x}$ and $\bar{y}$ are the means of the respective variables. Generally, the PC value lies in between $[1, 1]$ if the value is –1 then the variables are negatively correlated otherwise the variables are positively correlated. In case the value is 0, then there is no correlation between the variables. In [65], a correlation-based filter algorithm is developed to heuristically search an attribute subset through a correlation-based evaluation metric:

$$Merit_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \tag{2.24}$$

where S is the feature subset, $\overline{r_{cf}}$ is the average attribute-class correlation, and $\overline{r_{ff}}$ is the attribute-attribute inter-correlation. Subsequently, Lei and Liu utilized an entropy-based method to measure the correlation between each pair of attributes [69]. To estimate the relevance and redundancy, symmetrical uncertainty (SU) is used:

$$SU(X, Y) = 2\left[\frac{IG(X|Y)}{H(X) + H(Y)}\right], \tag{2.25}$$

where $IG(X|Y)$ is the information gain of variable X conditioned on variable Y:

$$IG(X|Y) = H(X) - H(X|Y), \tag{2.26}$$

where $H(X)$ is the entropy of X:

$$H(X) = -\sum_i P(x_i) \log P(x_i), \tag{2.27}$$

and $H(X|Y)$ is the joint entropy between X and Y:

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_i) \log P(x_i|y_i), \tag{2.28}$$

where $P(x_i)$ is the prior probabilities for all values of X, and $P(x_i|y_i)$ is the posterior probabilities of X given the values of Y. Thus, the correlation between any feature F_i and class variable C is called C-correlation denoting $SU_{i,c}$, and the correlation between each pair of feature F_i and F_j $(i \neq j)$ is called F-correlation denoting $SU_{i,j}$. Finally, the optimal feature subset contains all strong relevant attributes and weak relevant but non-redundant attributes. In [70], a C4.5 decision tree is used to select the features. Specifically, the features that appeared in the top three levels of a pruned decision tree on the dataset are selected as the candidate. Then, the final feature subset is formed by a union of all the attributes from the 5 rounds of the above process.

Mutual information is another widely used evaluation measurement in feature selection [66, 67, 71, 72]. Given two random variables X and Y, their mutual information is defined in terms of their probabilistic density functions $P(x)$, $P(y)$, and $P(x,y)$:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \left(\frac{P(x,y)}{P(x)P(y)} \right). \tag{2.29}$$

In [67], a minimal-redundancy-maximal-relevance (mRMR) criterion is introduced to find the optimal feature subset. In Max-Relevance, the selected feature x_i has the largest mutual information $I(x_i;c)$ with the target class variable c, reflecting the largest dependency on the target class. In terms of sequential search, the m best individual features, i.e., the top m features in the descent ordering of $I(x_i;c)$, are often selected to derive the optimal feature subset. Specifically, Max-Relevance is to search features satisfying the following equation:

$$\max D(S,c), \quad D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i;c), \tag{2.30}$$

where S is the feature subset, c is the class variable, which is measured by the mean value of all mutual information values between individual feature x_i and class c. By selecting the feature only using Max-Relevance, it is likely that the selected features could have rich redundancy, *i.e.*, the dependency among these features could be large. If two features highly depend on each other, it would not help to improve the respective class-discriminative power. Therefore, the following minimal redundancy (Min-Redundancy) condition can be added to select exclusive features:

$$\min R(S), \quad R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j). \tag{2.31}$$

The criterion combining the above two criteria is called "minimal-redundancy-maximal-relevance" (mRMR). The combined evaluation criterion $\Phi(D,R)$ is defined as:

$$\max \Phi(D,R), \Phi = D - R. \tag{2.32}$$

However, mRMR only minimizes mutual information between features and ignores the class-dependent information, which might be influenced by the selected features [72]. To estimate the redundancy more accurately, some conditional mutual information-based methods have been developed by estimating the redundancy between features conditioned on class varibales [66, 71]. Therefore, the redundancy is estimated by:

$$\min R(S), \quad R = \frac{1}{|S|^2} \sum_{x_i \in S} \sum_{x_j \in S} I(x_i, x_j) - I(x_i, x_j|C). \tag{2.33}$$

2.2.4.2 Wrapper-based Methods

Wrapper methods take the classification error or accuracy rate as the feature evaluation standard. Compared with filter methods, the wrapper ones could achieve higher classification accuracy and tend to have a smaller subset size, however, it has poor generalization capability and high time complexity [72, 73]. The most important part is to design an objective function and deploy an optimization algorithm to solve it. In [59], a Selective Bayesian classifier (SBC) is developed to iteratively select an attribute from the whole space of attributes by maximizing the classification accuracy of naive Bayes. More specifically, the attribute subset is initialized as an empty set. Then, a forward greedy search algorithm is utilized to add the best attribute into the subset at a time, in which the performance gain of NB is maximized. Finally, the optimal attribute subset is derived iteratively until no more improvement on classification accuracy. However, the greedy search in SBC often results in the local optimum [58]. Hence, Jiang *et al.* introduced a random selection strategy to derive the optimal attribute subset through a gradient descent optimization algorithm, by either maximizing the conditional log-likelihood or minimizing the mean squared error [58]. At first, the candidate feature subset is determined by using the performance of naive Bayes measured by classification accuracy (ACC), the area under curve (AUC) and conditional log likelihood (CLL). Then, the optimal number of features in the subset is derived by:

$$o = \log_2 p + 1, \tag{2.34}$$

where p is the number of features in the candidate feature subsets. Subsequently, the o best features are randomly selected from the whole space of features in which the performance is improved at most. To speed up the selection process, Bermejo *et al.* combined the naive Bayes classifier with the incremental wrapper feature selection algorithm [74].

Recently in [60], feature selection models are constructed based on a trivial extension of each other. The features are ranked by using mutual information with class variables in descending order. Then, a set of naive Bayes models are constructed by adding each feature into the subset from the sorted feature set. Consequently, the objective function is defined as:

$$s^* = \underset{s \in \{1,2,\ldots,m\}}{\arg\min} \sqrt{\frac{1}{n} \sum_{\boldsymbol{x} \in \mathcal{D}_{train}} (1 - p(y|\boldsymbol{x})_s)^2}, \tag{2.35}$$

where s is the number of features selected from the sorted feature set $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_a]$, n is the number of training samples and $p(y|\boldsymbol{x})_s$ is the posterior probability using the first s features in $\boldsymbol{F}$. The optimal s^* is determined by maximizing the posterior in which the first s^8 features are derived as the optimal feature subset.

2.2.5 Feature-weighting Naive Bayes

Instead of directly removing the redundant feature in feature selection, feature weighting methods alleviate the independence assumption of naive Bayes by assigning different weights to the features to enhance the discrimination power [19, 21, 22, 32–36]. Feature selection can be viewed as a special case of feature weighting by assigning a weight of 0 or 1 to the feature. In feature-weighted naive Bayes, the MAP is defined as:

$$\hat{c}(\boldsymbol{t}) = \underset{c \in \boldsymbol{C}}{\arg\max}\, \hat{P}(c|\boldsymbol{t})^{\boldsymbol{w}}, \tag{2.36}$$

where $\boldsymbol{w}$ is the weight vector the instance $\boldsymbol{t}$. To define the weight of each feature, many feature-weighting methods can be divided into filter-based methods and wrapper-based methods. The former often utilizes statistical measurements or heuristics to decide the weight, otherwise, the latter utilizes the classification feedback to optimize the weight. In general, the feature weight is normalized in $[0, 1]$.

2.2.5.1 Filter-based Methods

Filter-based methods [19, 32–36] utilize the characteristics of the data to determine attribute weights, *e.g.*, gain ratio, the minimum depth in the decision tree, Kullback-Leibler (KL) divergence and mutual information. In [37], a gain ratio-based feature weighting method is designed to determine the feature importance in which a feature with a higher gain ratio deserves a larger weight. Therefore, the weight of each feature is derived by using the average gain ratio across all features:

$$w_i = \frac{GR(A_i)}{\frac{1}{m}\sum_{i=1}^{m} GR(A_i)}, \tag{2.37}$$

where m is the number of features, $GR(A_i)$ is the gain ratio of using feature A_i to partition the given training instances, which is estimated by simply using the following equation:

$$GR(A_i) = \frac{I(A_i; C)}{H(A_i)}, \tag{2.38}$$

where $I(A_i; C)$ is the mutual information between feature A_i and class variable C defined in Eqn. 2.29 and $H(A_i)$ is the entropy defined in Eqn. 4.2.

Hall proposed a decision tree-based feature weighting (DTFW) method which defined the weights by utilizing the minimum depth in a decision tree [33]. Specifically, the feature weight is inversely proportional to the minimum depth at which it is tested in the built unpruned decision tree, and then the estimated weights are stabilized by averaging across 10 decision trees learned on 50% of the training data. Consequently, the weight is derived by:

$$w_i = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{\sqrt{d_{ti}}}, \tag{2.39}$$

where d_{ti} is the minimum depth at which feature A_i is tested in the built unpruned decision tree t, and T is the total number of the built decision trees. If a feature does not appear in the tree, the corresponding weight is set to 0, and the depth of the root node is set to 1 initially.

Lee *et al.* determined the weights by using the Kullback-Leibler (KL) divergence between attributes and class labels [34]. For each feature value a_i in ith feature A_i, it could provide the discriminative information with respect to the class variable C. Hence, the Kullback-Leibler (KL) divergence is used to estimate the amount of information: $KL(C|a_i) = \sum\limits_{c \in C} \log \frac{P(c|a_i)}{P(c)}$. Then, the weight for the feature A_i is estimated by averaging the KL measures of all feature values in A_i:

$$\begin{aligned} w_i &= \frac{1}{Z}\sum_{a_i} P(a_i)KL(C|a_i) \\ &= \frac{1}{Z}\sum_{a_i} P(a_i)\sum_{c} P(c|a_i)\log\frac{P(c|a_i)}{P(c)} \\ &= \frac{1}{Z}\sum_{a_i}\sum_{c} P(a_i)P(c|a_i)\log\frac{P(c|a_i)}{P(c)} \\ &= \frac{1}{Z}\sum_{a_i}\sum_{c} P(a_i,c)\log\frac{P(a_i,c)}{P(a_i)P(c)}, \end{aligned} \tag{2.40}$$

where $Z = \frac{1}{m}\sum_{i=1}^{m} w_i$ is a normalization constant to constrain the weight in $[0, 1]$.

In [32], a deep feature weighting method (DFW) is developed by using a correlation-based feature selection filter [65]. At first, the best feature subset is determined by using symmetrical uncertainty defined in Eqn. 2.25. Then, a simple and effective feature-weighting method is designed:

$$w_i = \begin{cases} 2 & if\ A_i\ is\ selected, \\ 1 & otherwise, \end{cases} \tag{2.41}$$

where the feature in the selected subset is weighted by 2, otherwise it is weighted by 1. Different from other feature-weighting methods, the learned feature weights are not only used to estimate the posterior probability but also the likelihood probability. The developed DFW is successfully applied in the state-of-the-art naive Bayes classifiers for text classification, *e.g.*, multinomial naive Bayes (MNB) [75].

Recently, Jiang *et al.* developed a correlation-based attribute-weighting NB, which defines the weight of each attribute as a sigmoid transformation of the difference between mutual relevance and average mutual redundancy [19]. They argue that the highly predictive features should be highly correlated with the class and uncorrelated with other features similar to the max-relevancy-min-redundancy (mRMR) in [67]. Based on this premise, the weight is defined as proportional to the difference between the feature-class correlation and the average feature-feature intercorrelation by using mutual information:

$$d_i = \underbrace{NI(A_i;C)}_{relevance} - \underbrace{\frac{1}{m-1}\sum_{j=1 \wedge j\neq i}^{m} NI(A_i;A_j)}_{average\ redundancy}, \tag{2.42}$$

where $NI(A_i;C)$ is the normalized mutual inforamtion between A_i and C representing the relevance:

$$NI(A_i;C) = \frac{I(A_i;C)}{\frac{1}{m}\sum_{i=1}^{m} I(A_i;C)}. \tag{2.43}$$

and $NI(A_i;A_j)$ is normalized mutual information between A_i and A_j representing the redundancy:

$$NI(A_i;A_j) = \frac{I(A_i;A_j)}{\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j=1 \wedge j\neq i}^{m} I(A_i;A_j)}. \tag{2.44}$$

where m is the number of feature dimensions. The resulting value of difference D_i may be negative which is not suitable for measuring the feature importance. Thus, the weight for ith feature is transformed into the range $(0,1)$ by using the logistic sigmoid function:

$$w_i = \frac{1}{1+e^{-d_i}}. \tag{2.45}$$

Recently, Zhang *et al.* recently developed a weighted naive Bayes method combining instance weights with attribute weights (AIWNB) [20]. In the feature weighting stage, the weight for each feature is estimated by using the measurement of the difference between the mutual relevance and the average mutual redundancy used in [19]. Then, two instance weighting methods are designed for eager learning and lazy learning ways. For eager learning, each instance weight is estimated by using the frequency of the attribute value across the whole training set. The detailed description can refer to the weighting method in [14]. Since eager learning has a high computational cost in the training phase, the lazy learning method directly computes the weight in the testing phase. The weight for ith training instance $\boldsymbol{x}_i$ with respect to the testing instance t is defined as:

$$w_i^{ins} = 1 + s(\boldsymbol{x}_i, \boldsymbol{t}), \tag{2.46}$$

where $s(\boldsymbol{x}_i, \boldsymbol{t})$ is the similarity derived using Eqn. 2.21. By combining the feature weights with instance weights in two different ways, the resulting AIWNB^E and AIWNB^L are introduced. AIWNB has achieved better trade-offs between generalization ability and discrimination power and hence results in superior classification performance compared with other state-of-the-art naive Bayes classifiers. The feature weighting filters rely on heuristic measurements for feature importance. They often have high efficiency but no guarantee for the optimal solution.

2.2.5.2 Wrapper-based Methods

In addition to the filter-based methods, wrapper-based methods often achieve higher classification performance by iteratively optimizing attribute weights. Due to the iterative process, wrapper-based methods usually have higher time complexity. In [37], Zhang and Sheng updated attribute weights based on a hill-climbing strategy to maximize AUC. Each feature weight w_i is first initialized as 1. Then, w_i is updated by:

$$w_i(k) \leftarrow w_i(k-1) + \triangle w(k), \tag{2.47}$$

where k is the number of iterations and $\triangle w(k)$ is the incremental weight in nth iteration and defined as:

$$\triangle w(k) = \eta O(auc)(1 - O(auc))^2, \tag{2.48}$$

where η is the learning rate, auc is the current value of AUC, and $O(auc)$ is defined as:

$$O(auc) = \frac{1}{1 + e^{-auc}}. \tag{2.49}$$

Once the improvement on AUC is less than a small value, the optimization process is terminated.

Wu and Cai developed a differential evolution-based feature weighting wrapper for the naive Bayes classifier, which utilizes a differential evolution search to optimize feature weights by maximizing the classification accuracy of the learned model [41]. Firstly, a population of attribute weight vectors is randomly generated in which the weights are constrained to be between 0 and 1. Then, the differential evolution processes mutation, crossover and selection to evolve the population. To effectively find the optimal weight vector, a fitness function is defined to determine if a mutation can replace the current weight vector with a new one. Then, a greedy search strategy is employed to select a weight vector from mutated ones as offspring only if the fitness function is better than that of the target one, otherwise, the target is maintained in the next iteration. The fitness function is defined as follows:

$$F(\boldsymbol{w}) = \frac{\sum_{i=1}^{n}(P_i^{\boldsymbol{w}} - P_i + 1)}{n}, \tag{2.50}$$

where n is the number of training instances, $P_i^{\boldsymbol{w}}$ is the posterior probability for ith instance and P_i is the ground-truth posterior probability. Once the optimal weight vector is derived, it is used to make the predictions on the testing data.

In [21], Zaidi *et al.* developed a class-independent attribute weighting method called WANBIA proposed to iteratively optimize attribute weights by minimizing the mean squared error between predicted and ground-truth labels or maximizing the conditional log-likelihood posterior probability. The posterior function is hence re-defined as:

$$\hat{P}(c|\boldsymbol{x})^{\boldsymbol{w}} = \frac{\pi_c \prod_j \theta_{c,j}^{w_j}}{\sum_{c'} \pi_{c'} \prod_j \theta_{c',j}^{w_j}}, \tag{2.51}$$

where π_c is the prior probability that sample $\boldsymbol{x}$ belongs to class c and $\theta_{c,j}$ is the likelihood of the jth attribute of $\boldsymbol{x}$ given the class c estimated from training samples using Eqn. 3.4, $\boldsymbol{w} = [w_1, w_2, \ldots, w_m]$ is the weight vector and w_j is the weight of the jth attribute. Then, the Conditional Log-Likelihood (CLL) function is defined as,

$$CLL(\boldsymbol{w}) = \sum_{i=1}^{n} \log \hat{P}(c|\boldsymbol{x}_i)^{\boldsymbol{w}}. \tag{2.52}$$

By maximizing the CLL objective function, the optimal weight vector can be derived through a gradient descent optimization algorithm. Instead of maximizing the supervised posterior, one can also minimize Mean Square Error (MSE) between the estimated posterior and the posterior derived from the ground-truth label:

$$MSE(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{c\in C}\left(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})^{\boldsymbol{w}}\right)^2, \tag{2.53}$$

where $P(c|\boldsymbol{x})$ is defined as:

$$P(c|\boldsymbol{x}_i) = \begin{cases} 1 & if\ c = c_i, \\ 0 & otherwise. \end{cases} \tag{2.54}$$

By minimizing the MSE objective function, the derived optimal weight vector can achieve a similar performance as maximizing CLL objective function. All the feature weights are initialized as 0 and optimized via the above two objective functions.

Very recently, Jiang *et al.* developed CAWNB [22], which determines the optimal weight for each attribute of different classes to capture more characteristics of the dataset, instead of ignoring the class dependency as in [21]. Hence it achieves excellent classification performance on many benchmark datasets. The weight matrix on a pre-class basis is defined as:

$$\boldsymbol{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{l,1} & w_{l,2} & \cdots & w_{l,m} \end{bmatrix}$$

where l is the number of classes and $w_{c,j}$ is the weight of the jth attribute for class c. Then, the posterior function is defined as:

$$\hat{P}(c|\boldsymbol{x})^{\boldsymbol{W}} = \frac{\pi_c \prod_j \theta_{c,j}^{w_{c,j}}}{\sum_{c'} \pi_{c'} \prod_j \theta_{c',j}^{w_{c',j}}}, \tag{2.55}$$

Similarly, CAWNB utilized the same objective function to derive the optimal weight matrix. Unlike WANBIA [21], which assigns the same attribute weight for all classes, CAWNB [22] assigns different weights to different classes, so that the CAWNB model is more complicated and more prone to over-fitting, especially when the dataset is small and WANBIA performs better on generalization performance.

In recent years, some embedded weighted naive Bayes methods have been also developed. In [42], Yu *et al.* developed a hybrid attribute-weighting method by initializing the weights through a correlation-based filter and then adjusting the weights through a wrapper. The initial weights are computed by using the correlation measure in [19].

Then, the weight vector is optimized by maximizing the following objective function:

$$f(\boldsymbol{w}) = \delta(\hat{c}(\boldsymbol{x}_i), c(\boldsymbol{x}_i)), \tag{2.56}$$

where $\hat{c}(\boldsymbol{x}_i)$ is the predicted class label for $\boldsymbol{x}_i$ derived using Eqn. 3.10 and $c(\boldsymbol{x}_i)$ is the ground-truth label.

2.3 Data Discretization

Discretization methods, as one of the basic reduction techniques, have been efficiently and effectively deployed in classification algorithms especially for NB classifiers [9, 21, 22]. Assuming a dataset S consisting of n examples, m attributes, and c class labels, a discretization scheme $\mathcal{D}_A$ for continuous attribute A can be generated, which divides this attribute into k discrete intervals:

$$\{[d_0, d_1], (d_1, d_2], ..., (d_{k-1}, d_k]\}, \tag{2.57}$$

where d_0 and d_k is the minimum and maximal value of attribute A. And the set of cut points of A can be represented by:

$$\mathcal{P}_A = \{d_1, d_2, ..., d_{k-1}\}. \tag{2.58}$$

There are four main steps in the discretization process shown in Fig. 3.1. Firstly, the continuous attribute is sorted to be discretized. Then, the evaluation process is designed to select the cut point for splitting or merging according to correlation, gain or classification performance. Thirdly, the interval can be either split or merge depending on the search strategies. Finally, the stopping criterion is specified to stop the discretization process with the trade-off between a lower number of intervals, good comprehension, and consistency. Discretization methods have been deployed to extract knowledge from data in many machine learning algorithms such as decision tree [77, 78], rule-based learning [79] and naive Bayes [9, 20, 22]. Discretization methods can be categorized according to many properties [76]. 1). Local vs. Global. Local methods [44, 80] generate intervals

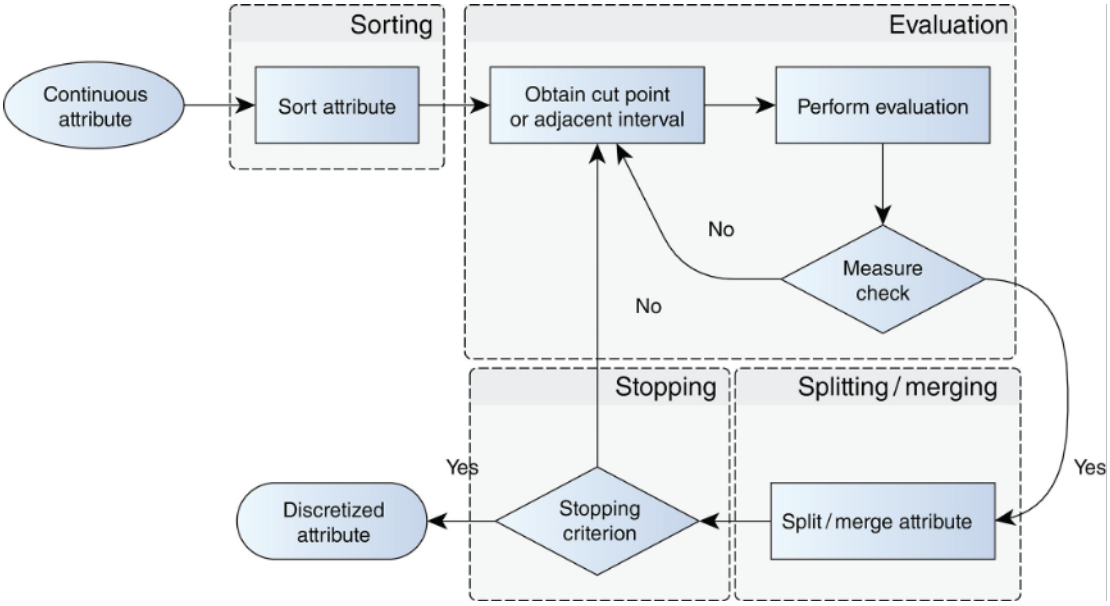


Figure 2.3: The main steps of discretization process [76].

based on partial data, whereas global ones [23–25, 45] consider all available data. 2). Dynamic vs. Static. Dynamic discretizers [81] interact with learning models whereas static ones [24, 82] execute before the learning stage. 3). Splitting vs. Merging. This relates to the top-down split [23, 25, 80] or bottom-up merge [82] strategy in producing new intervals. 4). Univariate vs. Multivariate. Univariate algorithms [24, 78, 80] discretize each attribute separately whereas multivariate discretizers [45, 83] consider a combination of attributes when discretizing data. 5). Direct vs. Incremental. Direct methods [45, 84] divide the range into several intervals simultaneously, while incremental ones [23–25, 44, 80] begin with a simple discretization and improve it gradually by using more criteria. The popular discretization methods are shown in Table 3.2.

Depending on whether the class label is used, discretization methods can be divided into supervised, semi-supervised and unsupervised methods [76]. Unsupervised methods include equal-width and equal-frequency discretization [78]. Semi-supervised methods are comparatively less studied. One of the representative methods is MODL, which derives the discretization scheme by applying the Bayesian rule on both labeled and unlabeled data [127]. Supervised methods can be further divided into wrapper-based methods [76, 83, 128] and filter-based methods [23–25, 80, 82]. The former optimizes the discretization scheme by utilizing the classification feedback [76, 83, 128], while the latter optimizes some indirect target for data discretization, *e.g.*, information entropy [44, 80], mutual information [81] and interdependency [23–25].

Wrapper-based methods [45, 83, 128, 129] often iteratively derive the optimal discretization scheme by using the classification results as the feedback signal. Among them, evolutionary algorithms are often utilized to select a set of cut points to discretize data by maximizing the classification accuracy and minimizing the number of intervals [45]. Recently, Tahan and Asadi developed an evolutionary multi-objective discretization to handle the imbalanced datasets [83]. To reduce the search space during discretization, Tran *et al.* firstly initializes the discretization scheme by using the MDLP criterion and utilizes barebones particle swarm optimization to fine-tune the derived scheme [129]. In [128], the particle swarm optimization strategy is used to explore the interaction between features to better discretize the data. To handle high-resolution satellite remote sensing, Chen *et al.* developed a genetic algorithm based on the fuzzy rough set to effectively explore the data association [130].

Filter-based methods [23, 25, 80, 82, 131] have a strong theoretical background and have gained popularity in recent years. MDLP discretizer has been widely used in many classifiers [9, 20, 22]. It is one of the most popular top-down discretization methods,

Table 2.2: The list of discretization methods.

Acronym	Ref.	Acronym	Ref.	Acronym	Ref.
EqualWidth	[85]	EqualFrequency	[85]	Chou91	[86]
D2	[87]	ChiMerge	[88]	1R	[89]
ID3	[90]	MDLP	[44]	CADD	[91]
MDL-Disc	[92]	Bayesian	[93]	Friedman96	[94]
ClusterAnalysis	[95]	Zeta	[96]	Distance	[97]
Chi2	[98]	CM-NFD	[99]	FUSINTER	[100]
MVD	[101]	Modified Chi2	[102]	USD	[103]
Khiops	[104]	CAIM	[23]	Extended Chit	[105]
Heter-Disc	[106]	UCPD	[107]	MODL	[108]
ITPF	[109]	HellingerBD	[110]	DIBD	[111]
IDD	[112]	CACC	[25]	Ameva	[113]
Unification	[114]	PKID	[115]	FFD	[115]
CACM	[116]	DRDS	[117]	EDISC	[118]
U-LBG	[119]	MAD	[120]	IDF	[121]
IDW	[121]	NCAIC	[122]	Sang14	[123]
IPD	[124]	SMDNS	[125]	TD4C	[126]
EMD	[45]	EMDID	[83]		

which hierarchically partitions the data to maximize the information entropy [44]. To avoid excessive splitting, it defines a stop criterion derived from the theory of channel coding. Xun *et al.* developed a multi-scale discretization method to obtain the set of cut points with different granularity and utilized the MDLP criterion to determine the best cut point [80]. Apart from entropy, other statistical measures have also been widely deployed in data discretization [24, 78, 82]. Kurgan and Cios developed a CAIM criterion based on a quanta matrix to select boundary points iteratively within a predefined number of intervals [23]. Recently, Cano *et al.* extended CAIM to discretize the multi-label data [24]. Tsai *et al.* introduced a discretization method based on CACC by taking the overall data distribution into account [25]. In [132], low-frequency values are discretized and the correlation between discrete attribute and continuous attribute is used to guide the discretization process. Chi-square statistics such as Modified Chi2 and extended Chi2 have been recently used to discretize data [82], which measure the relationship between the discretized attribute and the classification variable.

Most discretization methods [23–25, 44, 80] emphasize maximizing the discriminant power, but they pay little attention to the generalization capability, *e.g.*, they often restrict the number of discrete intervals to be small, in the hope of achieving a satisfactory generalization ability. If a discretization method considers maximizing the discriminant power and the generalization ability simultaneously, the subsequent classifier will achieve a better classification performance on novel testing data.

2.3.1 Representative Discretization Methods

2.3.1.1 Unsupervised Methods

Among them, there are some representative discretization methods. In an unsupervised manner, the popular dicretization methods include equal frequency (EF) [85], equal width (EW) [85], proportional k-interval discretization (PKID) [84] and fixed frequency discretization (FFD) [84]. For both equal-frequency and equal-width discretization, the minimum and maximum values of the continuous attribute are first identified. Then, the equal-frequency algorithm sorts all values in ascending order and divides the range into a user-defined number of intervals so that every interval contains the same number of samples, The equal-width discretization then divides the range into the user-defined number of intervals in which all the intervals have the same width. Instead of specifying the number of intervals generated, Fixed frequency discretization divide a range of an attribute into a set of intervals so that each interval shares the user-defined frequency [115]. To find an appropriate trade-off between the bias and variance of the probability estimation, PKID is introduced to adjust the number and size of intervals to the number of training instances [84], where the number of intervals k is defined as,

$$k = \sqrt{n}, \tag{2.59}$$

where n is the number of training instances. For each interval, the number of instances in each interval is also set to k. PKID often achieves superior classification performance without user input compared with other unsupervised discretization methods.

2.3.1.2 Supervised Methods

Entropy-based Methods

For supervised discretization methods, information theory is widely used because of its strong theoretical background [23, 44, 67, 81, 133, 134]. In discretization, Minimum Description Length Principle (MDLP) discretizer is one of the most important top-down methods by developing an entropy-based selection criterion. To evaluate all boundary points, the class entropy of the partitions is derived as an evaluation measure. The objective is to minimize the class entropy to select the best cut point for each binary partition. Finally, MDLP is used to define the stopping criterion, a cut point T will be accepted iff:

$$Gain(A,T;S) > \frac{log_2(N-1)}{N} + \frac{\Delta(A,T;S)}{N}, \tag{2.60}$$

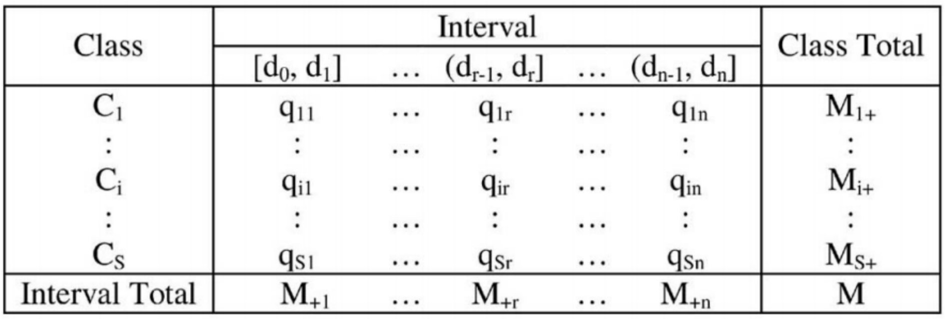
where $Gain(A,T;S)$ describes the information gain of a cut point T, which divided the current example set S into two subsets S_1 and S_2 given the attribute A. $Gain(A,T;S)$ is calculated by:

$$Gain(A,T;S) = Ent(S) - \frac{|S_1|}{|S|}Ent(S_1) + \frac{|S_2|}{|S|}Ent(S_1), \tag{2.61}$$

and

$$\Delta(A, T; S) = log_2(3^k - 2) - [kEnt(S) - k_1Ent(S_1) - k_2Ent(S_2)], \tag{2.62}$$

where $Ent(S)$ is class entropy defined in [44]. Other entropy-based discretization methods includes ID3 [90], FUSINTER [100] and Gini index [114].



Class	Interval					Class Total
	$[d_0, d_1]$	…	$(d_{r-1}, d_r]$	…	$(d_{n-1}, d_n]$	
C_1	q_{11}	…	q_{1r}	…	q_{1n}	M_{1+}
:	:	…	:	…	:	:
C_i	q_{i1}	…	q_{ir}	…	q_{in}	M_{i+}
:	:	…	:	…	:	:
C_S	q_{S1}	…	q_{Sr}	…	q_{Sn}	M_{S+}
Interval Total	M_{+1}	…	M_{+r}	…	M_{+n}	M

Figure 2.4: Quanta Matrix.

Statistical-based Methods

Statistical-based discretization is another representative algorithm, which evaluates the cut point by measurement of dependency or correlation among features [23, 25, 88, 91, 96, 98, 108]. In statistical discretization methods, a two-dimensional frequency matrix (called quanta matrix) is often used to measure the relationship between discretization scheme and class variables as described in Fig. 3.3. In Fig. 3.3, S is the number of classes and n is a number of candidate cut points. q_{ir} is the total number of continuous values distributed in the interval $(d_{r-1}, d_r]$ given the i-th class. $Mi+$ is the total number of samples with the i-th class and M_{+r} is total number of continuous values of attribute A distributed the interval $(d_{r-1}, d_r]$. The probability of the occurrence that attribute A values are within the interval $D_r = (d_{r-1}, d_r]$ given the class C_i, can be estimated by:

$$p_{ir} = p(C_i, D_r|A) = \frac{q_{ir}}{M}. \tag{2.63}$$

Then, the prior probability that attribute A values given class C_i can be estimated by:

$$P_{i+} = p(C_i) = \frac{M_{i+}}{M}. \tag{2.64}$$

The probability of each interval that attribute A values are distributed interval $D_r = (d_{r-1}, d_r]$ is defined as follow:

$$P_{+r} = p(D_r|A) = \frac{M_{+r}}{M}. \tag{2.65}$$

The mutual information between discretization variable D for attribute A and class C is defined as:

$$I(C, D|A) = \sum_{i=1}^{S}\sum_{r=1}^{n} p_{ir} \log_2 \frac{p_{ir}}{p_{i+}p{+r}}. \tag{2.66}$$

Similarly, the class-attribute information and Shannon's entropy are defined, respectively, as follows:

$$INFO(C, D|A) = \sum_{i=1}^{S}\sum_{r=1}^{n} p_{ir} \log_2 \frac{p{+r}}{p_{ir}}, \tag{2.67}$$

$$H(C, D|A) = \sum_{i=1}^{S}\sum_{r=1}^{n} p_{ir} \log_2 \frac{1}{p_{ir}}. \tag{2.68}$$

Then, the class-attribute dependent discretizer (CADD) [91] defined a CAIR criterion to select the cut points:

$$CAIR(C, D|A) = \frac{I(C, D|A)}{H(C, D|A)}. \tag{2.69}$$

The CAIR criterion is used to measure the interdependence between classes and the discretized attribute that the larger CAIR values mean the better correlation between class labels and the discrete intervals. However, it has some drawbacks: 1). it initializes the discretization by the user-defined number of intervals; 2). The maximum entropy is used for initialization which may mislead the cut point selection by using the CAIR criterion. Therefore, the CAIM algorithm was developed to discretize an attribute into the smallest number of intervals and maximize the class-attribute interdependency and then improve the classification performance. To measure the class-attribute interdependency, the CAIM criterion is defined as:

$$CAIM(C, D|A) = \frac{\sum_{r=1}^{n} \frac{max_r^2}{M_{+r}}}{n}. \tag{2.70}$$

Although CAIM outperforms the other discretization algorithms because of its efficiency and performance gain given to classification algorithms, it has two limitations: 1). CAIM always generates a simple discretization scheme with few intervals by assigning a high factor to the number of generated intervals when it discretizes an attribute; 2). CAIM only considers the class with the most samples and ignores all the other target classes which would decrease the quality of the generated discretization scheme. To address the problem, class-attribute contingency coefficient (CACC) discretization was developed.

Inspired by the contingency coefficient, CACC can generate a better discretization scheme and lead to the improvement of classification performance. Generally, the contingency coefficient is used to measure the strength of dependence between variables. Given the quanta matrix, the selection criterion can be defined as:

$$C = \sqrt{\frac{y}{y+M}}, \tag{2.71}$$

where $y = M[(\sum_{i=1}^{S} \sum_{r=1}^{n} \frac{q_{ir}^2}{M_{i+}M_{+r}}) - 1]$. It's obvious that the contingency coefficient takes the distribution of all samples into account by using all the values in the quanta matrix. To reduce the time complexity and prevent the over-fitting problem, y in the contingency coefficient is divided by $log(n)$. Thus, the CACC criterion is defined as:

$$CACC = \sqrt{\frac{y}{y+M}}, \tag{2.72}$$

where $y = M[(\sum_{i=1}^{S} \sum_{r=1}^{n} \frac{q_{ir}^2}{M_{i+}M_{+r}}) - 1]/\log(n)$.

Chi^2-based Methods

The Chi^2-based methods are also famous supervised discretization methods in a bottom-up manner based on statistical independence including ChiMerge [88], Chi2 [135], Modified Chi2 [102] and Extended Chi2 [105]. The chi-square (χ^2) statistic is used to determine whether the current interval pair is to be merged or not. The χ^2 test is a statistical technique used to test the association between a variable and its category. The χ^2 statistic measures the degree of similarity between neighboring intervals at a certain level of significance. Intuitively, two intervals tend to be merged into one interval if they are statistically similar measured by χ^2.

The ChiMerge algorithm is the earliest Chi^2-based discretization method [88]. In the algorithm, each distinct value of a continuous variable is assumed as an independent interval, and then χ^2 statistic is tested for whether the adjacent intervals are to be merged or not. If the χ^2 statistic for adjacent intervals is smaller than the predefined χ^2 threshold, adjacent intervals are merged because they are assumed statistically similar. The χ^2 value is defined as:

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(I_{ij} - E_{ij})^2}{E_{ij}}, \tag{2.73}$$

where $m = 2$ is the number interval to be compared, k is the number of classes, E_{ij} is the expected frequency of I_{ij} which is defined as,

$$I_{ij} = \frac{R_i * C_j}{N}, \tag{2.74}$$

where R_i is the number of samples in $i$the interval, C_j is the number of samples belonging to class c and N is the total number of samples. The discretization process of ChiMerge starts with sorting the numerical features for each pair of adjacent and then its intervals are continuously merged until a termination condition is met. The pair of adjacent values which has the lowest χ^2 value are merged into one interval. Merging continues until all pairs of intervals have χ^2 values exceeding the parameter χ^2 threshold and is used as a stopping criterion. The χ^2 threshold is determined by selecting a desired significance level (α).

Chi2 discretization is an extension of ChiMerge which automatically discretization without user input by introducing an inconsistency rate as the stopping condition [135]. Chi2 selects the appropriate level of statistical significance and combines neighboring intervals until the inconsistency rate is met,

$$InConCheck(data) > \delta, \tag{2.75}$$

where δ is a pre-defined value indicating the level of inconsistency. The inconsistency rate is the sum of all the inconsistency counts divided by the total number of instances. For all the matching instances (without considering their class labels), the inconsistency count is the number of the instances minus the largest number of the instances of the class labels; for example, there are n matching instances, and among them, c_1 instances belong to label 1, c_2 to label 2, and c_3 to label 3 where $c1 + c2 + c3 = n$. If c_3 is the largest among the three, the inconsistency count is $(n - c_3)$.

Modified Chi2 is an improved modification of Chi2, which fixed the over-merging problem of Chi2 [102]. To precisely measure the inconsistency, modified Chi2 determines the level of inconsistency by using rough set theory,

$$L_c = \frac{\sum |\underline{B}X_i|}{|U|}, \tag{2.76}$$

where U is the set of all objects of the data, X can be any subset of U $(X \subset U)$, $\underline{B}X_i$ is the lower approximation of X in B $(B \subseteq A)$, A is the set of attributes, X is a classification of U $(i \in \{1, 2, \ldots, n\})$. Hence, in the modified Chi2 algorithm, inconsistency checking $(InConCheck() < \delta)$ of the original Chi2 algorithm was replaced by maintaining the level of consistency L_c after each step of discretization $(L_c{-}discretized \leq L_c{-}original)$. By using this inconsistency rate as the stopping criterion, it guaranteed that the fidelity of the training data could be maintained to be the same after discretization. In addition, it made the discretization process completely automatic.

Extended Chi2 is yet another improvement of Chi2 which has the ability to deal with uncertain data [105]. Instead of using the inconsistency rate to determine the merging process, extended Chi2 utilized the least upper bound of data misclassification error $\xi(C, D)$ to guide the discretization process,

$$\xi(C, D) = max(m_1, m_2), \tag{2.77}$$

where C is the equivalence relation set, D is the decision set, and $C^* = \{E_1, E_2, \ldots, E_n\}$ is the equivalence class, m_1 and m_2 are defined as:

$$\begin{aligned} m_1 &= 1 - min\{c(E, D) | E \in C^* and\ 0.5 < c(E, D)\}, \\ m_2 &= max\{c(E, D) | E \in C^* and\ c(E, D) < 0.5\}, \\ c(E, D) &= 1 - \frac{card(E \bigcap D)}{card(E)}, \end{aligned}$$

where *card* denotes set cardinality. Chi^2-based methods are often effective in handling non-linear relationships between the attribute and the class variable.

Wrapper-based Methods

Recently, the wrapper-based discretization methods have achieved significant performance improvement on classification tasks. For example, the evolutionary cut point selection has successfully deployed in multivariate discretization [45]. Ramirez *et al.* presented an evolutionary multivariate discretizer (EMD), which selects a set of boundary cut points to generate the discrete intervals by minimizing the classification error. To select the most appropriate discretization scheme from the data population, a fitness function with two objectives, minimizing classification error and the number of cut points, is defined as:

$$Fitness(Q) = \alpha\frac{|Q|}{|BP|} + (1-\alpha)\Delta, \tag{2.78}$$

where Q is a subset of cut points selected from the initial set with all potential boundary points BP, Δ represents the classification error based on discretized data and α is an input parameter to balance classification error and the number of intervals.

To address the low generalization of discretization on imbalanced data, Tahan and Asadi developed a multi-objective discretization wrapper that derives the optimal scheme by maximizing AUC, minimizing the number of cut points and maximizing low-frequency cut points [83]. The first objective is described as follows,

$$f_1(S_j) = 1 - \frac{AUC_{F-CT} + AUC_{F-KNN}}{2}, \tag{2.79}$$

where AUC_{F-CT} and AUC_{F-KNN} represent the area under the ROC curve estimated by the CART and KNN classifiers, respectively, S_j is the discretization scheme for jth individual of the population. Compared with classification accuracy, AUC remains indifferent to the imbalanced distribution and hence yields solutions with adequate classification ability. The second objective is to minimize the number of cut points in the current chromosome,

$$f_2(S_j) = |S_j|. \tag{2.80}$$

Besides the above two objectives, the frequency of the chosen cut points also plays an important role to minimize information loss,

$$f_3(S_j) = \sum_{i=1}^{C_i} freq_i, \tag{2.81}$$

where C_i is the $i$the cut point. Wrapper-based discretization methods often select a set of cut points that result in better classification performance but at a high computational cost.

2.3.2 Evaluation Criteria

When comparing different discretization methods, there are serval criteria to evaluate the relative strength and weaknesses of each algorithm including the number of intervals, inconsistency, predictive classification rate and time complexity. Firstly, a continuous attribute should be discretized into discrete ones with as few values as possible to make sure learning effectiveness and efficiency. Secondly, inconsistency is associated with the number of different classes in the same discrete values. The desired inconsistency level of a discretization approach should be 0. Thirdly, a well-designed discretization method is able to reduce classification errors. Finally, the time complexity of discretization is very important for real-time applications and the discretization process should be performed efficiently.

In conclusion, most discretization methods aim to find a reasonable discretization that can achieve a better trade-off between the number of intervals and information gain. Subsequently, classification algorithms can have balanced generalization capability and discrimination power. Due to insufficient information gain, MDLP often leads to an early stop so that too few discrete intervals are generated, and hence leads to a huge loss in discriminative information. Similarly in CAIM, the discretization scheme is too simple to provide enough discrimination power to classifiers. CACC tried to address the problem by considering the distribution of all samples. Since many recent discretization methods optimize the discretization scheme based on evolutionary algorithms by minimizing the classification error [45, 83, 136], information-based methods are highly overlooked. Besides, existing researchers rarely consider discretization based on labeled data while neglecting the amount of unlabeled data in real-world applications.

2.4 Data Augmentation

Data augmentation techniques have been widely used in image classification [137–139], text classification [140], and signal processing [141] They can be broadly categorized into instance augmentation methods [137–139, 141–144] and feature augmentation methods [46–48, 145–147]. The former augments more training samples from existing ones to effectively reduce the gap between the training set and the testing set, which improves the generalization ability of the model. Comparatively, feature augmentation methods are less studied, which enrich the discriminant information of the original feature space by augmenting new features so that the discrimination power of the classification model is enhanced [47, 145].

2.4.1 Instance-space Augmentation

Instance augmentation methods are often utilized to enlarge the dataset, which can be further categorized as model-free methods [137] and model-based methods [139, 142–144]. Model-free augmentation methods directly transform an existing instance to a new one by a set of transformation techniques, *e.g.*, image random erasing [137], image rotation, scaling, cropping and flipping [138], and text editing in text classification [148]. In [137], random erasing is developed to make the model robust to occlusion by masking off a randomly selected region in an image. In text classification, a set of text editing techniques such as random insertion, deletion, replacement, and swap are employed to expand the training set [148]. Model-based augmentation methods employ deep learning models to generate new instances, *e.g.*, generative adversarial networks (GNNs) [142] and convolutional neural networks (CNNs) [139, 143, 144]. Moreno-Barea *et al.* utilized a set of GAN-based methods to generate the synthetic samples for improving the classification accuracy on small datasets [142]. In [139], the deep feature vectors extracted by CNNs are augmented by randomly adding the difference vectors extracted from a small set of clean and occluded image pairs to enhance the classification performance on occluded images. With the expanded size and increasing variety of the training dataset, instance augmentation could often effectively improve the generalization ability of classification models [138].

The aforementioned instance augmentation methods focus on augmenting the unstructured data, *e.g.*, text and image. For structured data, traditional methods are commonly used to address classification problems where the data is imbalanced in which the dataset has one or multiple minority classes. For an imbalanced dataset, the learned model easily overfits the data with the majority class and hence generalizes poorly on the data with the minority class. To alleviate this problem, oversampling algorithms are widely used to synthesize data points for minority class [149]. More specifically, these algorithms create synthetic instances to balance the class distributions of the original dataset by utilizing contextual information [150]. To create these new instances, oversampling methods use linear or geometric interpolations between a randomly selected observation and one of its neighboring instances [150–152].

Synthetic minority over-sampling technique (SMOTE) is one of the most commonly used algorithms for oversampling [150–152]. SMOTE creates synthetic data points along a line connecting a randomly chosen under-represented class instance and one of its closest neighbors [150]. Specifically, a sample from the minority class is randomly selected and then its k-nearest neighbors are computed. For each nearest neighbor, new synthetic samples are generated along the line segment between the two samples. This process is repeated until the desired balance between the minority and majority classes is achieved. However, SMOTE algorithm can lead to overfitting if the synthetic samples are too similar to the original minority class samples. In addition, it may not work well if the minority class samples are not well separated from the majority class samples. Hence, there have been modifications to address these problems, *e.g.*, Borderline-SMOTE [151] and Geometric-SMOTE [152].

Borderline-SMOTE is an adaptation of SMOTE that generates synthetic samples only for the minority class examples that are close to the decision boundary between the minority and majority classes [151]. Intuitively, it's not necessary to augment data for the minority class samples that are already well separated from the majority class, while the ones that are close to the boundary may benefit from them. Geometric-SMOTE is another variation of the SMOTE algorithm designed to generate synthetic samples for the minority class by considering the geometric structure of the feature space [152]. The target of Geometric-SMOTE is to generate synthetic samples by interpolating not only between pairs of minority class samples, but also between triplets of minority class samples that form a triangle in the feature space, which allows Geometric-SMOTE to capture more complex structures and relationships within the minority class. Geometric-SMOTE often achieves superior performance especially those with complex geometric structures in the feature space.

2.4.2 Feature-space Augmentation

Traditionally, feature dimensionalities are often transformed into low-dimensional space with most discriminant information by using feature extraction methods, *e.g.*, principal component analysis (PCA) [153, 154] and linear discriminant analysis (LDA) [26]. However, both PCA and LDA do not consider the dependence between features which may lead to the loss of discriminant power. Recently, many feature-space augmentation methods have been developed to enrich the discriminant information of data [47, 139, 145, 155, 156]. Wang *et al.* developed an adaptive feature augmentation scheme for intrusion detection framework via logarithm marginal density ratios transformation based support vector machine [145]. Chen*et al.* introduce a camera correlation aware feature augmentation method for a person re-identification system to capture the correlation information across different camera views [156]. In [47], a kNN-based feature augmentation method is designed to enrich the original feature space and hence improve the discriminant power of the multi-dimensional classification model. In literature, both feature-space augmentation and instance-space augmentation methods are less explored in the naive Bayes classifier. Since the instance-space augmentation method may introduce biases in the data [157], Feature-space augmentation is more robust to boost the discriminant power of naive Bayes.

In recent years, feature augmentation methods have been proven to be another effective way to boost the performance of classification tasks by incorporating new features to the original ones, which can also be divided into model-free methods [47, 145, 146] and model-based methods [46, 48, 147]. The former ones utilize a set of transformation techniques to generate new features and then add them with the original ones [47, 145, 146]. Wang *et al.* utilize the logarithm marginal density ratios transformation to capture the feature correlations into the augmented features and hence improve the performance of intrusion detection [145]. In [47], feature relationships among neighboring instances are exploited to produce the new feature vectors and enhance the multi-dimensional classification performance. These methods often have limited performance improvements without the learning mechanism [48]. The latter ones learn features automatically by a set of learning architectures, *e.g.*, CNN with self-attention mechanism [147] and artificial neural network [46, 48]. In [147], the convolutional feature maps extracted by CNNs are augmented with self-attentional feature maps to capture both local and global information for improving the performance of vision tasks. Recently, Li *et al.* first utilized relative transformation to model the relationships among classes of data samples and then employed an artificial neural network to generate the augmented features for enhancing the discriminant power of classifiers [48]. By exploiting the intrinsic data residing in the original feature space, feature augmentation methods could effectively derive new well-pose features to enhance the discrimination power of subsequent classification tasks.

2.4.3 Summary

In the Bayesian classification framework, naive Bayes has achieved excellent performance due to its simplicity and efficiency [1–3, 5–7]. However, the independence assumption in naive Bayes often does not hold so that numerous improved naive Bayes methods have been developed to alleviate this problem [11–13, 17, 19–22]. Among them, the wrapper-based methods often achieve the state-of-the-art classification performance [21, 22]. For example in WANBIA, the attributes are weighted on a class-independent basis in which each weight is assigned to each attribute ignoring the class variable [21]. In CAWNB, attributes of different classes are weighted differently to enhance the discrimination power of the model. CAWNB better captures the characteristics of the dataset and achieves significant performance improvements compared with other attribute-weighting methods. However, with more weights to be optimized, the model complexity increases and hence over-fitting may occur, especially if the dataset is small. To alleviate the problem, we propose a regularized naive Bayes to automatically balance the generalization ability and discrimination power. The work is discussed in Chapter 3.

To handle the mixed data types in the dataset, naive Bayes often relies on the discretization method to first transform the numerical attributes into discrete ones and hence the probability distribution can be better estimated [9, 20, 22]. Most discretization methods [23–25, 44, 80] emphasize maximizing the discriminant power, but they pay little attention to the generalization capability, *e.g.*, they often restrict the number of discrete intervals to be small, in the hope of achieving a satisfactory generalization ability. Thus, we first propose a semi-supervised discretization framework with an adaptive discriminative discretization criterion to enhance the discrimination power of naive Bayes classifiers as described in Chapter 4. In Chapter 5, we further explore the well-designed selection criterion to derive an optimal discretization scheme and hence lead to the better trade-off between generalization ability and discrimination power of classifiers.

Due to the independence assumption, the discrimination power of naive Bayes is limited in two ways: 1) lacking a mechanism to model the correlations between features; 2) ignoring the local data structure formed by jointly considering all the feature dimensions of neighboring samples. To address these two problems, many augmentation techniques are developed to augment original data in instance-space [150, 152] or feautre-space [46–48]. Since naive Bayes is not sensitive to instance size, feature augmentation is a more effective way to enhance the discrimination power of naive Bayes classifiers. In Chapter 6, we propose a feature augmentation framework for naive Bayes classifiers to boost their discriminant power.

Chapter 3

A Regularized Attribute Weighting Framework for Naive Bayes

The Bayesian classification framework has been widely used in many fields, but the covariance matrix is usually difficult to estimate reliably[1]. To alleviate the problem, many naive Bayes (NB) approaches with good performance have been developed. However, the assumption of conditional independence between attributes in NB rarely holds in reality. Various attribute-weighting schemes have been developed to address this problem. Among them, class-specific attribute weighted naive Bayes (CAWNB) has recently achieved good performance by using classification feedback to optimize the attribute weights of each class. However, the derived model may be over-fitted to the training dataset, especially when the dataset is insufficient to train a model with good generalization performance. This paper proposes a regularization technique to improve the generalization capability of CAWNB, which could well balance the trade-off between discrimination power and generalization capability. More specifically, by introducing the regularization term, the proposed method, namely regularized naive Bayes (RNB), could

[1] This work has been published in IEEE Access.

well capture the data characteristics when the dataset is large, and exhibit good generalization performance when the dataset is small. RNB is compared with the state-of-the-art naive Bayes methods. Experiments on 33 machine-learning benchmark datasets demonstrate that RNB outperforms the compared methods significantly.

3.1 Introduction

The Bayesian classification framework is fundamental to statistical pattern recognition and widely deployed in many machine-learning tasks [158–163]. Bayesian decision rule with 0/1 loss function leads to the optimal classification in statistical pattern recognition [164]. However, the estimated covariance matrix in Bayesian classification often deviates from the data population due to the curse of dimensionality, which may reduce classification performance [164]. To tackle the problem, many naive Bayes (NB) approaches [49–51, 165] have been developed, which regularize the covariance matrix to a diagonal matrix. In these methods, it is assumed that each feature dimension is conditionally independent, and then the posterior probability can be estimated separately for each feature dimension. NB classifiers are competitive with many latest classifiers as shown in [166, 167].

However, NB may be oversimplified as the assumption of strong independence is often invalid, resulting in a decrease in classification performance [168]. Many improved naive Bayes classifiers have been developed to alleviate the conditional independence assumption, which can be broadly divided into five categories: 1) Structure extension [11, 12]; 2) Instance selection [13, 15]; 3) Instance weighting [43]; 4) Feature selection [17, 18]; 5) Feature weighting [5, 19, 21, 22, 32–42]. Among these methods, attribute-weighting methods [5, 19, 21, 22, 32–42] relieve the independence assumption by assigning different weights to different attributes so that the discriminative features will have a larger weight.

Attribute-weighting methods can be further divided into filter-based methods [19, 32–36] and wrapper-based methods [5, 21, 22, 37–42]. The former determines the attribute weights in advance by using the general characteristics of the data, while the latter determines the attribute weights by using classification feedback to minimize the classification error. In most cases, the filter-based methods calculate weights faster than the wrapper-based ones, but the classification accuracy of the latter is higher than that of the former.

Attribute-weighting methods often assign the same weight to each attribute in different classes, e.g. Zaidi *et al.* weighed the attributes to alleviate naive Bayes' independence assumption (WANBIA) [21]. In class-specific attribute weighted naive Bayes (CAWNB) [22], attributes of different classes are weighted differently to enhance the discrimination power of the model. CAWNB better captures the characteristics of dataset and achieves significant performance improvements compared with other attribute-weighting methods. However, with more weights to be optimized, the model complexity increases and hence over-fitting may occur, especially if the dataset is small. To alleviate the problem, we propose to add a regularization term to the formulation of CAWNB to penalize the model complexity, which will tend to use simpler models to avoid over-fitting, similarly as in [154, 164, 169].

Naive Bayes can be regarded as a regularized form of the Bayesian classification framework by restricting the covariance matrix to be diagonal [164]. L1- or L2-regularization has been widely used in machine-learning tasks [170, 171]. L2-regularization [171] could be applied on the model parameters to encourage the attribute weights with poor effect to decay towards zero and assign higher weights to attributes with higher effect. Alternatively, L1-regularization could be applied to the model parameters of CAWNB, which is more robust to noise and outliers than L2-regularization. L1-regularization in general produces better results, but at a higher computational cost [170]. Sparse representation is an example of L1-regularization [170].

Both L1-regularization and L2-regularization will introduce a significant computational overhead. In this paper, a simple yet effective way is proposed to regularize CAWNB, i.e. add a simpler model to constrain CAWNB. Simpler models usually achieve better generalization performance [172]. WANBIA is simpler than CAWNB, as the number of weights estimated in WANBIA are fewer than that in CAWNB. Hence, it will improve the generalization capability of CAWNB by integrating with the simpler model WANBIA. Furthermore, it will not significantly increase the computational complexity by integrating these two models, as both share similar procedures to solve the optimization problem [21, 22]. The proposed approach is named as regularized naive Bayes (RNB).

In the proposed RNB, the target is to find the optimal model parameters $\boldsymbol{M} =\{\boldsymbol{W}, \boldsymbol{w}, \alpha\}$ to minimize the difference between the posterior derived from the ground-truth label and the posterior $P(\boldsymbol{M})$ estimated from the data, where

$$P(\boldsymbol{M}) = \alpha P_D(\boldsymbol{W}) + (1-\alpha)P_I(\boldsymbol{w}). \tag{3.1}$$

$P_D(\boldsymbol{W})$ is the posterior probability with attributes weighted on a per-class basis, and $\boldsymbol{W}$ is the matrix to weigh the attributes differently for different classes. $P_I(\boldsymbol{w})$ is the posterior probability with attributes weighted the same for all classes, and $\boldsymbol{w}$ is the weight vector for the attributes. $P_D(\boldsymbol{W})$ is a more complex model than $P_I(\boldsymbol{w})$, as more weights need to be optimized in $\boldsymbol{W}$ than that in $\boldsymbol{w}$. Thus, $P_I(\boldsymbol{w})$ is a simpler model that can provide better generalization capabilities.

Now the challenge is how to jointly find the optimal model parameters including $\boldsymbol{W}$, $\boldsymbol{w}$, and α. To achieve this, a gradient-based optimization procedure is proposed, similar to L-BFGS-M [173] used in CAWNB and WANBIA. More specifically, the partial derivatives of $P(\boldsymbol{M})$ w.r.t. $\boldsymbol{W}$, $\boldsymbol{w}$ and α are derived, and a gradient-descent-based method is utilized to iteratively update $\boldsymbol{W}$, $\boldsymbol{w}$ and α respectively, towards the objective of minimizing the classification error. Compared with other regularization methods, the proposed method requires minimal modifications to the optimization problem of CAWNB, and it does not significantly increase the computational complexity.

In the proposed formulation, α is used to automatically adjust the trade-off between discrimination power and generalization capability. More specifically, when the dataset is small and hence a simpler model is preferred, α will be smaller and hence a larger weight will be assigned to $P_I(\boldsymbol{w})$, which will ensure better generalization capabilities. This is verified by the experiments shown in Section 3.4.

To validate the effectiveness of the proposed RNB, a series of empirical comparisons have been conducted with state-of-the-art naive Bayes on the collection of 33 benchmark classification datasets from the University of California at Irvine (UCI) repository [174]. Experimental results show that the performance of RNB is significantly better than all compared methods [18, 19, 21, 22, 32, 41, 42, 165].

The contributions of this paper are summarized as follows: 1) The poor generalization capability of CAWNB is identified and RNB is proposed to address the problem. 2) An optimization procedure is designed to derive the optimal model of the proposed RNB. 3) The proposed RNB improves the generalization performance of previous methods and automatically balances the discrimination power and the generalization capability, so that better performance can be obtained regardless of the size of datasets.

The rest of the paper is organized as follows. Section 3.2 reviews related work. Then, the proposed regularized naive Bayes is introduced in section 3.3. In section 3.4, experimental comparisons with state-of-the-art naive Bayes are conducted to demonstrate the effectiveness of the proposed method. Finally, this work is concluded in section 3.5.

3.2 Related Work

Naive Bayes classifiers have been widely used in many applications [49–51]. As the strong assumption of feature independence in NB is often invalid, many improvements have been developed, which can be broadly divided into 5 categories. The first category, structure extension [11, 12], extends the structure of naive Bayes to represent the feature dependencies. The second category, instance selection [13, 15], employs the principle of local learning to build a set of local naive Bayes classifiers using a subset of the dataset. The third category, instance weighting [43], weights the instances differently in order to maximize the discriminant power. The fourth category, feature selection [17, 18], removes the strongly correlated or irrelevant features, as those features are harmful to reliable classification, and/or selects the most discriminative feature subset. The fifth category, weighted naive Bayes, tackles the problem by assigning different weights to attributes so that the discriminative features have a larger weight and hence the discriminative power will increase [5, 19, 21, 22, 32–42]. The attribute-weighting methods can be further categorized into filter-based methods [19, 32–36] and wrapper-based methods [5, 21, 22, 37–42].

Filter-based methods [19, 32–36] utilize the characteristics of the data to determine attribute weights. Lee *et al.* determined the weights by using the Kullback-Leibler (KL) divergence between attributes and class labels [34]. In [33], Hall defined the weights by utilizing the minimum depth in a decision tree. In [32], the conditional probabilities of naive Bayes are estimated by deeply computing feature weighted frequencies. Recently, Jiang *et al.* developed a correlation-based attribute-weighting NB, which defines the weight of each attribute as a sigmoid transformation of the difference between mutual relevance and average mutual redundancy [19]. Filter-based approaches determine the weights in advance by measuring the relationship between features and classification variables, such as mutual information, KL divergence and correlation.

Wrapper-based methods iteratively utilize the classification feedback to optimize attribute weights. Due to the iterative process, wrapper-based methods usually have higher time complexity and better classification performance than filter-based ones. In [37], Zhang and Sheng updated attribute weights based on a hill-climbing strategy to maximize the classification accuracy. Wu and Cai utilized a differential evolution algorithm to determine the weights [41]. In [42], Yu *et al.* developed a hybrid attribute-weighting method by initializing the weights through a correlation-based filter and then adjusting the weights through a wrapper. Zaidi *et al.* optimized attribute weights by minimizing the mean squared error between predicted and ground-truth labels [21]. Very recently, Jiang *et al.* developed CAWNB [22], which determines the optimal weight for each attribute of different classes to capture more characteristics of the dataset, instead of ignoring the class dependency as in [21]. Hence it achieves excellent classification performance on many benchmark datasets.

Unlike WANBIA [21], which assigns the same attribute weight for all classes, CAWNB [22] assigns different weights to different classes, so that the CAWNB model is more complicated and more prone to over-fitting, especially when the dataset is small. Some form of regularization to CAWNB is required to improve its generalization performance.

3.3 Regularized attribute-weighted Naive Bayes

3.3.1 Problem Analysis of Previous Naive Bayes Methods

In the Bayesian classification framework, the posterior probability is defined as:

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c)P(c)}{P(\boldsymbol{x})}, \tag{3.2}$$

where $\boldsymbol{x}$ is the feature vector and c is the classification variable. Because it is difficult to reliably estimate the likelihood $P(\boldsymbol{x}|c)$ due to the curse of dimensionality, in naive Bayes methods, the likelihood is estimated by assuming that the attributes are independent given the classification variable c, which results in the following formulation:

$$P(\boldsymbol{x}|c) = \prod_{j=1}^{m} P(x_j|c), \tag{3.3}$$

where x_j is the j-th dimension of the feature vector $\boldsymbol{x}$, and m is the feature dimensionality. Then, the posterior probability can be estimated by:

$$P(c|\boldsymbol{x}) = \frac{P(c)\prod\limits_{j=1}^{m} P(x_j|c)}{\sum_{c'} P(c')\prod\limits_{j=1}^{m} P(x_j|c')}. \tag{3.4}$$

Naive Bayes regularizes the Bayesian framework by assuming that each attribute is independent conditioned on the classification variable, but this assumption is often invalid. To alleviate the problem, weights are assigned to attributes in WANBIA [21], and the weights are optimized via minimizing the mean squared error between the estimated posteriors and the posteriors derived using ground-truth labels.

Jiang *et al.* showed that attribute weighting should be class-specific to enhance the discrimination power of naive Bayes [22]. Thus, different weights are assigned to the attributes for different classes in CAWNB [22]. CAWNB is more complicated than WANBIA considering the number of model parameters. Class-specific attribute weights provide CAWNB with greater discrimination. However, the model complexity is considerably increased, so the generalization capability may decrease. The problem will be severe when the dataset is small, so the training samples are not enough to derive a reliable naive Bayes model.

To improve the generalization capability of CAWNB, we propose to add a simpler model, WANBIA, to constrain CAWNB. Besides, CAWNB is an improved version of WANBIA, and both share a similar optimization procedure. It will not significantly increase the computational complexity by integrating WANBIA into CAWNB.

3.3.2 Overview of Proposed Regularized Naive Bayes

In the proposed method, the target is to use the classification feedback to optimize the attribute weights. More precisely, the target is to find the optimal attribute weights to minimize the difference between the estimated posteriors and the posteriors derived from the ground-truth labels. The mean squared error is often used to capture such differences:

$$f = \frac{1}{2} \sum_{\boldsymbol{x}_i \in D} \sum_{c} (P(c|\boldsymbol{x}_i) - \hat{P}(c|\boldsymbol{x}_i))^2, \tag{3.5}$$

where D represents the whole dataset, $\hat{P}(c|\boldsymbol{x}_i)$ is the estimated posterior of class c given $\boldsymbol{x}_i$, and the posteriors derived from the ground-truth labels are defined as:

$$P(c|\boldsymbol{x}_i) = \begin{cases} 1 & if\ c = c_i, \\ 0 & otherwise. \end{cases} \tag{3.6}$$

The posterior $\hat{P}(c|\boldsymbol{x}_i)$ consists of two parts. The first part that emphasizes the discriminative power of the model, whose attributes are weighted on a class-dependent basis, is defined as:

$$\hat{P}_D(c|\boldsymbol{x}) = \frac{\pi_c \prod_j \theta_{c,j}^{w_{c,j}}}{\sum_{c'} \pi_{c'} \prod_j \theta_{c',j}^{w_{c',j}}}, \tag{3.7}$$

where $\boldsymbol{\pi} = [\pi_1,\ \pi_2, ...,\ \pi_l]$ are the prior probabilities, and π_c is the prior probability that sample $\boldsymbol{x}$ belongs to class c. The matrix $\boldsymbol{\Theta}$ of likelihood probabilities is defined as:

$$\boldsymbol{\Theta} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,m} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{l,1} & \theta_{l,2} & \cdots & \theta_{l,m}, \end{bmatrix}$$

where $\theta_{c,j}$ is the likelihood of the j-th attribute of $\boldsymbol{x}$ given the class c. $\boldsymbol{\pi}$ and $\boldsymbol{\Theta}$ are estimated from training samples using (3.13) and (3.14) respectively, as shown in section 3.3.3 later on.

$$\boldsymbol{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{l,1} & w_{l,2} & \cdots & w_{l,m} \end{bmatrix}$$

is the attribute-weighting matrix on a per-class basis and $w_{c,j}$ is the weight of the j-th attribute for class c.

The other posterior probability $\hat{P}_I(c|\boldsymbol{x})$ that emphasizes the generalization capability of the model, whose attributes are weighted on a class-independent basis, is defined as:

$$\hat{P}_I(c|\boldsymbol{x}) = \frac{\pi_c \prod_j \theta_{c,j}^{w_j}}{\sum_{c'} \pi_{c'} \prod_j \theta_{c',j}^{w_j}}, \tag{3.8}$$

where $\boldsymbol{w} = [w_1, w_2, \ldots, w_m]$ is the weight vector and w_j is the weight of the j-th attribute.

In the proposed RNB, the regularized posterior probability is defined as:

$$\hat{P}(c|\boldsymbol{x}) = \alpha \hat{P}_D(c|\boldsymbol{x}) + (1-\alpha)\hat{P}_I(c|\boldsymbol{x}), \tag{3.9}$$

where $\boldsymbol{M}$=$\{\boldsymbol{W},\ \boldsymbol{w},\ \alpha\}$ consists of class-dependent attribute weights $\boldsymbol{W}$, class-independent attribute weights $\boldsymbol{w}$ and a hyper-parameter α. α is used to balance the trade-off between the discrimination power and the generalization capability.

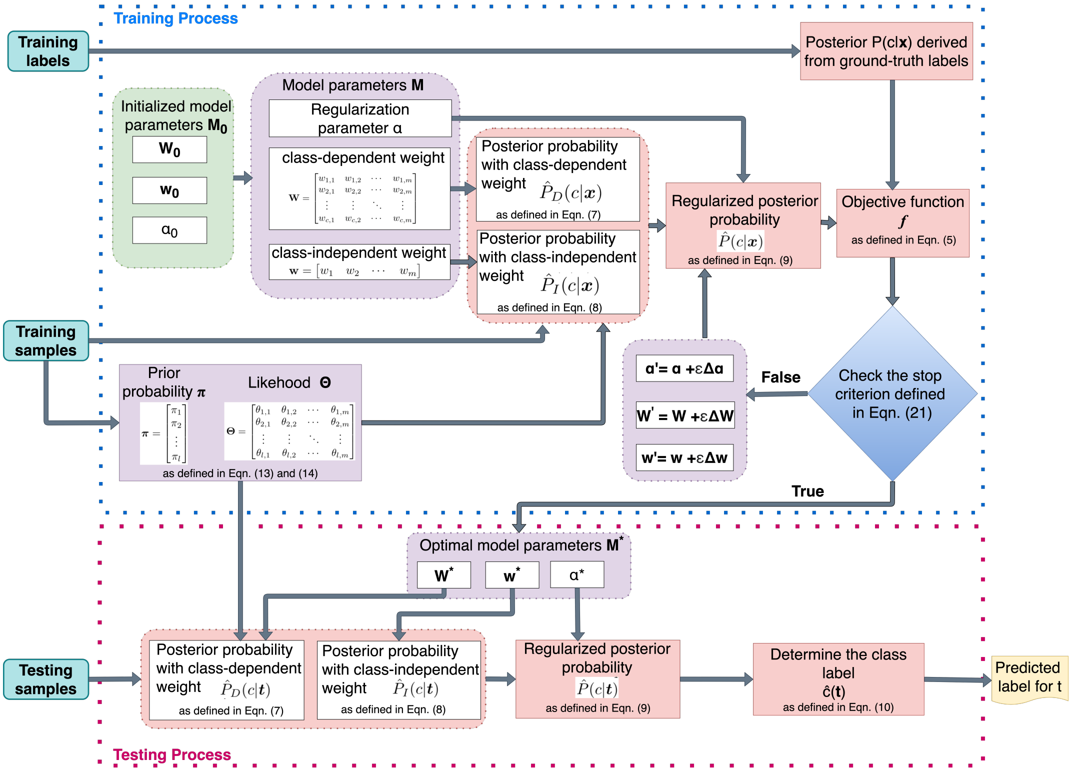

Training Process
Training labels
Posterior P(c|x) derived from ground-truth labels
Initialized model parameters M0
W0
w0
α0
Model parameters M
Regularization parameter α
class-dependent weight
$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{c,1} & w_{c,2} & \cdots & w_{c,m} \end{bmatrix}$
class-independent weight
$\mathbf{w} = [w_1 \quad w_2 \quad \cdots \quad w_m]$
Posterior probability with class-dependent weight $\hat{P}_D(c|\boldsymbol{x})$
as defined in Eqn. (7)
Posterior probability with class-independent weight $\hat{P}_I(c|\boldsymbol{x})$
as defined in Eqn. (8)
Regularized posterior probability $\hat{P}(c|\boldsymbol{x})$
as defined in Eqn. (9)
Objective function f
as defined in Eqn. (5)
Training samples
Prior probability π
Likehood Θ
$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_l \end{bmatrix}$
$\boldsymbol{\Theta} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,m} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{l,1} & \theta_{l,2} & \cdots & \theta_{l,m} \end{bmatrix}$
as defined in Eqn. (13) and (14)
α'= α +εΔα
W' = W +εΔW
w'= w +εΔw
False
Check the stop criterion defined in Eqn. (21)
True
Optimal model parameters M*
W*
w*
α*
Testing samples
Posterior probability with class-dependent weight $\hat{P}_D(c|\boldsymbol{t})$
as defined in Eqn. (7)
Posterior probability with class-independent weight $\hat{P}_I(c|\boldsymbol{t})$
as defined in Eqn. (8)
Regularized posterior probability $\hat{P}(c|\boldsymbol{t})$
as defined in Eqn. (9)
Determine the class label ĉ(t)
as defined in Eqn. (10)
Predicted label for t
Testing Process

Figure 3.1: Proposed regularized attribute weighting framework for naive Bayes.

The block diagram of the proposed regularized naive Bayes is shown in Fig. 3.1. In the training process, the elements in $\boldsymbol{W}$ and $\boldsymbol{w}$ are all initialized to 1 and α is initialized to 0.5, so that the initial model is the original naive Bayes. Then, $\hat{P}_D(c|\boldsymbol{x})$ and $\hat{P}_I(c|\boldsymbol{x})$ are estimated using training samples and these two posteriors are integrated as the regularized posterior $\hat{P}(c|\boldsymbol{x})$ with the weighting factor α, as shown in (3.9). Then, f is calculated as the sum of the squared differences between $P(c|\boldsymbol{x})$ and $\hat{P}(c|\boldsymbol{x})$, as shown in (3.5). The model parameters are optimized iteratively by using a gradient-descent-based method to minimize f until convergence. The detailed procedures to derive the optimal model parameters are given in Section 3.3.4. The class-independent weights significantly improve the generalization capability of the model, as evidenced in Section 3.4.

In the testing process, the estimated prior probabilities $\boldsymbol{\pi}$, the likelihood probabilities $\boldsymbol{\Theta}$ and the optimal model parameters $\boldsymbol{M}^*= \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$ are used to compute the posterior probability $\hat{P}(c|\boldsymbol{t})$ for a given test instance $\boldsymbol{t}$ by using (3.9). Finally, the class label of $\boldsymbol{t}$ is estimated by using MAP estimation as follows:

$$\hat{c}(\boldsymbol{t}) = \arg\max_{c \in \boldsymbol{C}} \hat{P}(c|\boldsymbol{t}), \tag{3.10}$$

where $\boldsymbol{C}$ is the set of labels for all classes.

3.3.3 Estimation of Prior Probabilities and Likelihood Probabilities

Firstly, prior probabilities $\boldsymbol{\pi}$ and likelihood probabilities $\boldsymbol{\Theta}$ are estimated based on training samples. Traditionally, the prior probability π_c for class c is estimated as follows:

$$\pi_c = \frac{\sum_{i=1}^{n} \delta(c_i, c)}{n}, \tag{3.11}$$

where n is the number of training samples, c_i is the class label of the i-th training instance, and $\delta(\bullet)$ is a binary function, which is 1 if its two parameters are identical and 0 otherwise. The likelihood function $\theta_{c,j}$ for the j-th attribute of class c is estimated as follows:

$$\theta_{c,j} = \frac{\sum_{i=1}^{n} \delta(x_{ij}, x_j)\delta(c_i, c)}{\sum_{i=1}^{n} \delta(c_i, c)}, \tag{3.12}$$

where x_{ij} is the j-th attribute value of the i-th training instance and x_j is the j-th attribute.

To make the estimation numerically stable, e.g. to avoid estimating π_c to 0 due to insufficient training samples, in the proposed method, the prior probability π_c and the likelihood $\theta_{c,j}$ are estimated by adding a regularization term as follow:

$$\pi_c = \frac{\sum_{i=1}^{n} \delta(c_i, c) + \frac{1}{l}}{n+1}, \tag{3.13}$$

$$\theta_{c,j} = \frac{\sum_{i=1}^{n} \delta(x_{ij}, x_j)\delta(c_i, c) + \frac{1}{n_j}}{\sum_{i=1}^{n} \delta(c_i, c) + 1}, \tag{3.14}$$

where n_j is the number of discretized values for the j-th attribute.

The aforementioned procedures work for discrete features. Continuous features are transformed into discrete features by using the Fayyad & Irani's MDL method [44].

Then, (3.13) and (3.14) are used to compute prior probabilities and likelihood probabilities of continuous features respectively in the same way as discrete ones.

3.3.4 Solving the Optimization Problem

Now the challenge is how to jointly find the optimal model parameters $\boldsymbol{M}$ including $\boldsymbol{W}$, $\boldsymbol{w}$, and α. To achieve this, a gradient-descent-based optimization procedure is proposed, similar to L-BFGS-M [173] used in CAWNB and WANBIA. More specifically, the target is to find the gradient direction of the objective function w.r.t. the model parameters $\boldsymbol{W}$, $\boldsymbol{w}$, and α, respectively. Then, the model parameters are updated iteratively along the gradient direction to minimize the error function defined in (3.5).

The partial derivative of f w.r.t. each element of $\boldsymbol{W}$, $w_{c,j}$, is given as follows:

$$\begin{aligned}\frac{\partial f}{\partial w_{c,j}} = -\alpha \sum_{\boldsymbol{x}\in D} &\left(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\right) \\ &\left[\hat{P}_D(c|\boldsymbol{x})(1-\hat{P}_D(c|\boldsymbol{x}))\log(\theta_{c,j})\right].\end{aligned} \tag{3.15}$$

Similarly, the partial derivative of f w.r.t. each element of $\boldsymbol{w}$, w_j is calculated as:

$$\begin{aligned}\frac{\partial f}{\partial w_j} = (\alpha-1)\sum_{\boldsymbol{x}\in D}\sum_{c} &\left(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\right)\hat{P}_I(c|\boldsymbol{x}) \\ &\left(log(\theta_{a_j|c}) - \sum_{c'}\hat{P}_I(c'|\boldsymbol{x})log(\theta_{c',j})\right).\end{aligned} \tag{3.16}$$

The detailed derivations are omitted here and a brief derivation is described in Appendix. Finally, the partial derivative of f w.r.t. α can be calculated as:

$$\frac{\partial f}{\partial \alpha} = \sum_{\boldsymbol{x}\in \boldsymbol{X}} \left(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x})\right)\left(\hat{P}_D(c|\boldsymbol{x}) - \hat{P}_I(c|\boldsymbol{x})\right). \tag{3.17}$$

After deriving the partial derivatives of the objective function f w.r.t. the model parameters, the model parameters $\boldsymbol{W}$, $\boldsymbol{w}$, and α are iteratively updated to minimize the classification error. After the i-th iteration of optimization, the model parameters

$\boldsymbol{W}_i, \boldsymbol{w}_i, \alpha_i$ are updated using the following equations:

$$\boldsymbol{W}_{i+1} = \boldsymbol{W}_i + \epsilon \nabla \boldsymbol{W}_i, \tag{3.18}$$

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \epsilon \nabla \boldsymbol{w}_i, \tag{3.19}$$

$$\alpha_{i+1} = \alpha_i + \epsilon \nabla \alpha_i, \tag{3.20}$$

where $\nabla \boldsymbol{W}_i$ is the gradient matrix whose elements are defined in (3.15), $\nabla \boldsymbol{w}_i$ is the gradient vector whose elements are defined in (3.16), $\nabla \alpha_i$ is the partial derivative defined in (3.17) and ϵ is the learning rate. The iteration will stop when:

$$\frac{f_i - f_{i+1}}{\max\left(|f_i|, |f_{i+1}|, 1\right)} < \eta, \tag{3.21}$$

where η is a predefined small constant. The optimal model is denoted as $\boldsymbol{M}^* = \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$.

The learning algorithms for training and testing are summarized in **Algorithm 1** and **Algorithm 2**, respectively.

Algorithm 1 Training algorithm

```
Input: $\boldsymbol{x}$: training samples, $f$: the objective function.
Output: the prior probabilities $\boldsymbol{\pi}$, the likelihood probabilities $\boldsymbol{\Theta}$, and the optimal model
    parameters $\boldsymbol{M}^*= \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$.
Estimate the prior probability $\pi_c$ using (3.13).
Estimate the likelihood probability $\theta_{c,j}$ using (3.14).
Derive the posterior probability $P(c|\boldsymbol{x})$ from the ground-truth labels using (3.6).
Initialize attribute weights of $\boldsymbol{W}$ and $\boldsymbol{w}$ to 1 and $\alpha$ to 0.5.
while stop condition (3.21) is NOT met do
    Derive the class-dependent posterior $\hat{P}_D(c|\boldsymbol{x})$ by (3.7).
    Derive the class-independent posterior $\hat{P}_I(c|\boldsymbol{x})$ by (3.8).
    Derive the regularized posterior $\hat{P}(c|\boldsymbol{x})$ by (3.9).
    Derive the objective function $f$ using (3.5).
    Derive the partial derivatives of $f$ w.r.t. $\boldsymbol{W}$, $\boldsymbol{w}$, $\alpha$ using (3.15), (3.16) and (3.17),
    respectively.
    Update $\boldsymbol{W}$, $\boldsymbol{w}$ and $\alpha$ using (3.18), (3.19) and (3.20), respectively.
end while
Return the prior probabilities $\boldsymbol{\pi}$, the likelihood probabilities $\boldsymbol{\Theta}$ and the optimal
model parameters $\boldsymbol{M}^* = \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$.
```

α is initialized to 0.5 so that the initial model will not bias the discrimination power or the generalization capability. α is optimized to achieve the best trade-off between

Algorithm 2 Testing algorithm

```
Input: $\boldsymbol{t}$: a test instance, $\boldsymbol{M}^*= \{\boldsymbol{W}^*, \boldsymbol{w}^*, \alpha^*\}$: the set of the optimal model parameters, $\boldsymbol{\pi}$: the prior probabilities, $\boldsymbol{\Theta}$: the likelihood probabilities.
Output: the class label of the test instance $\boldsymbol{t}$.
1: Derive the class-dependent posterior $\hat{P}_D(c|\boldsymbol{t})$ using (3.7).
2: Derive the class-independent posterior $\hat{P}_I(c|\boldsymbol{t})$ using (3.8).
3: Derive the regularized posterior $\hat{P}(c|\boldsymbol{t})$ using (3.9).
4: Determine the class label $\hat{c}(\boldsymbol{t})$ of the test instance $\boldsymbol{t}$ using (3.10).
5: Return the predicted class label $\hat{c}(\boldsymbol{t})$.
```

discrimination power and generalization capability. A small value of α means that a small weight is assigned to $\hat{P}_D(c|\boldsymbol{x})$, and a large weight is assigned to $\hat{P}_I(c|\boldsymbol{x})$. As a result, a better generalization capability is expected. Note that in the extreme case, the model is reduced to $\hat{P}_D(c|\boldsymbol{x})$ for $\alpha = 1$, or $\hat{P}_I(c|\boldsymbol{x})$ for $\alpha = 0$. All the weights of $\boldsymbol{W}$ and $\boldsymbol{w}$ are initialized to 1, which means that the model is initialized to naive Bayes at the beginning. In the proposed regularized naive Bayes, not only the prior probabilities and the likelihood probabilities are regularized to avoid numerical instability as shown in (3.13) and (3.14), but also the posterior is regularized to improve the generalization capability as shown in (3.9).

3.4 Experimental Results

The proposed approach is compared with original naive Bayes [175], Gaussian naive Bayes [165] and several state-of-the-art NB algorithms. TCSFS-NB improves the performance of naive Bayes through feature selection[18]. DAWNB [32] and CFW [19] are two recent filter-based attribute-weighting methods. The comparisons with them can illustrate the performance gain of the proposed RNB over filter-based approaches. DEAWNB [41], WANBIA [21], CAWNB [22] and CWANB [42] are four wrapper-based attribute-weighting methods in recent years. They can provide a comprehensive comparison to wrapper-based attribute-weighting methods. These competitors are summarized in Table 6.1.

Table 3.1: Description of competitors: original NB, Gaussian NB, one feature-selection-based method, two filter-based attribute-weighting methods and four wrapper-based attribute-weighting methods.

Algorithm	**Description**
NB [175]	Original naive Bayes method.
GNB [165]	Gassian naive Bayes method.
TCSFS-NB [18]	Test-cost-sensitive feature selection.
DAWNB [32]	Filter-based attribute weighting, with deep attribute weighting.
CFW [19]	Filter-based attribute weighting, with correlation-based attribute weighting.
DEAWNB [41]	Wrapper-based attribute weighting, with differential evolution-based attribute weighting.
WANBIA [21]	Wrapper-based attribute weighting, with attributes weighted in a class-independent manner.
CAWNB [22]	Wrapper-based attribute weighting, with attributes weighted in a class-specific manner.
CWANB [42]	Wrapper-based attribute weighting, with filter-based initialization and wrapper-based optimization for attribute weighting of each attribute.

3.4.1 Experimental Settings

Comprehensive experiments are conducted on a collection of 33 benchmark datasets from the UCI repository [2], which represent a wide range of domains and data characteristics [174]. Most datasets are from real-world problems such as diabetes, hepatitis and primary tumor, vehicle classification and letter recognition. Besides, the characteristics of the datasets including the number of instances, attributes and classes are significantly different. The sizes of datasets are between 57 and 20000, enough to evaluate how the algorithms perform on datasets of different sizes. For example, smaller datasets such as breast-cancer, heart-c and iris will prefer methods with better generalization capabilities. Attribute weighting methods with good discrimination power will perform better on larger datasets such as sick, hypothyroid, waveform-5000 and mushroom. In addition, 17 out of 33 datasets have missing values, which simulates the difficulties in real life when collecting datasets, and imposes additional challenges for classifiers. Besides numeric values, the attributes of some datasets are nominal values, which imposes

[2] These 33 datasets could be downloaded from "https://archive.ics.uci.edu/ml/index.php"

another challenge for classifier design. These 33 benchmark datasets provide a comprehensive evaluation of the effectiveness of the proposed RNB. The dataset descriptions are summarized in Table 3.2.

Table 3.2: 33 benchmark datasets are collected from real-world problems. The number of instances is widely distributed in 57 and 20000 which can provide a comprehensive evaluation on datasets of different sizes.

Dataset	Instance	Attributes	Classes	Missing values	Numeric values
anneal	898	39	6	Y	Y
audiology	226	70	24	Y	N
balance-scale	625	5	3	N	Y
breast-cancer	286	10	2	Y	N
breast-w	699	10	2	Y	N
colic	368	23	2	Y	Y
credit-a	690	16	2	Y	Y
credit-g	1000	21	2	N	Y
diabetes	768	9	2	N	Y
glass	214	10	7	N	Y
heart-c	303	14	5	Y	Y
heart-h	294	14	5	Y	Y
heart-statlog	270	14	2	N	Y
hepatitis	155	20	2	Y	Y
hypothyroid	3772	30	4	Y	Y
ionosphere	351	35	2	N	Y
iris	150	5	3	N	Y
kr-vs-kp	3196	37	2	N	N
labor	57	17	2	Y	Y
letter	20000	17	26	N	Y
lymphography	148	19	4	N	Y
mushroom	8124	23	2	Y	N
primary-tumor	339	18	21	Y	N
segment	2310	20	7	N	Y
sick	3772	30	2	Y	Y
sonar	208	61	2	N	Y
soybean	683	36	19	Y	N
splice	3190	62	3	N	N
vehicle	846	19	4	N	Y
vote	435	17	2	Y	N
vowel	990	14	11	N	Y
waveform-5000	5000	41	3	N	Y
zoo	101	18	7	N	Y

The missing values in the datasets are replaced with the average value of the numeric attributes or the mode of the nominal attributes in the available data. In CAWNB, they use Fayyad & Irani's MDL method [44] to discretize numeric attributes which may lead to information loss. Thus, in the experiments, the Fayyad & Irani's MDL method is fine-tuned to reduce the information loss. Besides, two irrelevant attributes are deleted, such as “instance name” in “splice” and “animal” in “zoo”.

The results of NB, DAWNB, DEAWNB, WANBIA and CAWNB are obtained from [22]. The results of TCSFS-NB, DAWNB and CWANB are obtained from [18], [32] and [42], respectively. GNB is implemented using Weka and the proposed RNB is implemented in MATLAB. The classification accuracy of the proposed algorithm on each dataset is derived via 10-fold cross-validation. During optimization, η is set to 10^{-7} in the stop criterion defined in (3.21). The learning rate ϵ is determined using the linear search programs [176].

3.4.2 Comparison to State-of-the-art

The comparisons to the state-of-the-art algorithms on the 33 datasets are shown in Table 3.3. The symbol $\bullet$ represents the statistically significant improvements achieved by the proposed regularized naive Bayes for a paired one-side t-test with the p=0.05 significance level. The average classification accuracy and the $Win/Tie/Loss$ on the 33 datasets for all the algorithms are summarized at the bottom of Table 3.3. The average classification accuracy over all the datasets can provide a straightforward comparison of their performance. Each entry of $W/T/L$ in the table indicates that the competitor wins on W datasets, ties on T datasets and loses on L datasets compared to the proposed RNB.

From Table 3.3, it is evident that the proposed Regularized Naive Bayes (RNB) achieves the highest average classification accuracy. Specifically, RNB outperforms the original Naive Bayes and Gaussian Naive Bayes by 2.34% and 6.15% on average, respectively. Furthermore, when compared to the filter-based approaches, such as DAWNB [32] and CFW [19], RNB demonstrates average improvements of 2.26% and 1.82%, respectively. Notably, RNB also surpasses the feature-selection-based TCSFS-NB [18] by 2.32% on average.

Table 3.3: Experimental results for RNB versus NB [175], DAWNB [32], DEAWNB [41], WANBIA [21], CAWNB [22], CWANB [42], GNB[165], TCSFS-NB[18] and CFW[19]. RNB achieves the best classification accuracy among all approaches.

Dataset	RNB	GNB [165]	TCSFS-NB [18]	CFW [19]	CWANB [42]	CAWNB [22]	NB [175]	DAWNB [32]	DEAWNB [41]	WANBIA [21]
anneal	99.22	86.30 ●	98.26 ●	98.50 ●	98.55	99.47	96.36 ●	97.45 ●	98.41 ●	98.69
audiology	80.08	71.24 ●	74.20	74.22	77.52	80.96	75.74	77.11	76.08	78.08
balance-scale	78.55	90.40	70.72 ●	73.76 ●	70.01 ●	71.08 ●	71.08 ●	71.99 ●	69.26 ●	71.08 ●
breast-cancer	70.25	72.03	71.10	72.46	71.28	69.78	72.32	71.50	70.46	71.35
breast-w	96.99	96.00	96.58	97.14	97.07	96.50	97.25	97.30	96.91	96.51
colic	83.42	77.45 ●	84.13	83.34	82.83	83.07	81.20 ●	82.93	82.55	83.72
credit-a	86.09	77.68 ●	85.93	86.99	86.26	86.14	86.17	86.49	86.81	86.23
credit-g	**78.60**	75.40 ●	74.11 ●	75.70 ●	75.47 ●	76.04 ●	75.40 ●	74.27 ●	75.08 ●	75.59 ●
diabetes	78.64	76.30 ●	78.15	78.01	78.37	78.67	77.88	78.70	77.85	78.48
glass	**80.01**	48.60 ●	74.40 ●	73.37 ●	74.72 ●	73.69 ●	74.20 ●	72.00 ●	75.32 ●	73.82 ●
heart-c	83.54	82.84	82.48	82.94	83.71	83.03	83.73	83.11	82.38	83.73
heart-h	82.32	82.99	80.73	83.82	82.66	83.41	84.43	84.05	81.61	84.39
heart-statlog	82.96	83.70	83.70	83.44	85.04	84.33	83.74	83.33	83.59	84.74
hepatitis	**89.83**	83.87 ●	86.99	85.95	86.02	86.66	85.05	84.80	86.66	86.61
hypothyroid	99.52	95.23 ●	99.07 ●	98.56 ●	99.47	99.60	98.74 ●	98.15 ●	99.31	99.37
ionosphere	91.80	82.62 ●	91.57	91.82	92.77	92.74	91.37	91.79	91.71	92.73
iris	**97.33**	96.00	95.33 ●	94.40 ●	94.60 ●	94.67 ●	94.33 ●	94.53 ●	94.13 ●	94.33 ●
kr-vs-kp	93.08	87.89 ●	94.09	93.58	94.38	95.20	87.81 ●	91.86 ●	94.11	93.92
labor	91.90	91.23	87.13 ●	92.10	94.60	92.63	93.83	93.57	94.63	95.60
letter	**76.62**	64.12 ●	74.61 ●	75.22 ●	75.25 ●	75.42 ●	74.67 ●	75.33 ●	75.21 ●	75.55 ●
lymphography	84.30	83.11	82.20	84.81	81.47	83.76	85.70	83.39	84.24	84.48
mushroom	99.96	95.83 ●	99.70 ●	99.19 ●	99.84 ●	99.96	98.03 ●	99.02 ●	99.89 ●	99.90 ●
primary-tumor	47.30	46.90	46.25	47.20	45.69	47.15	47.11	43.84	47.34	48.53
segment	**95.84**	80.22 ●	93.97 ●	93.47 ●	95.27	94.68 ●	92.91 ●	93.84 ●	95.09	95.24
sick	97.56	92.92 ●	97.21	97.36	97.44	97.54	97.07	96.86 ●	97.59	97.47
sonar	**91.90**	67.79 ●	80.55 ●	82.56 ●	82.71 ●	84.58 ●	84.96 ●	83.72 ●	84.10 ●	83.85 ●
soybean	94.00	92.09 ●	91.64 ●	93.66	93.79	94.31	93.53	93.35	93.71	93.75
splice	**96.39**	95.30 ●	95.09 ●	96.19	96.19	95.81	95.58 ●	96.05	95.84	96.28
vehicle	69.61	44.80 ●	66.60 ●	62.91 ●	68.32	70.33	62.64 ●	62.82 ●	66.30 ●	68.57
vote	95.87	90.11 ●	96.30	92.11 ●	95.15	95.77	90.30 ●	92.62 ●	95.35	95.52
vowel	**75.56**	63.74 ●	68.11 ●	68.84 ●	70.45 ●	69.07 ●	66.00 ●	67.45 ●	68.19 ●	68.19 ●
waveform-5000	**85.84**	80.00 ●	81.67 ●	83.11 ●	84.22 ●	85.56	80.76 ●	80.99 ●	83.80 ●	84.65 ●
zoo	**98.09**	95.05 ●	93.69 ●	95.96	96.15	95.95	95.75 ●	94.05 ●	95.45 ●	95.75 ●
AVERAGE	**86.45**	**80.30**	**84.13**	**84.63**	**85.07**	**85.38**	**84.11**	**84.19**	**84.82**	**85.35**
$W/T/L$	-	1/9/23	0/17/16	0/14/19	0/24/9	0/25/8	0/15/18	0/16/17	0/21/12	0/23/10

[1] ● indicates that statistically significant improvement is achieved by the proposed RNB with significance level $p = 0.05$.
[2] The bold value of classification accuracy means the proposed RNB performs best on the dataset.

In comparison to the previous best algorithm, CAWNB, the proposed RNB shows more than a 1% improvement in average classification accuracy across 33 datasets. The enhancements are particularly notable in certain datasets. For instance, RNB achieves classification accuracies that are over 5% higher than those obtained by CAWNB on datasets such as balance-scale, glass, sonar, and vowel. On smaller datasets like glass, iris, and sonar, RNB significantly outperforms CAWNB and other methods, demonstrating its superior generalization capability. Even on larger datasets such as segment and letter, RNB shows statistically significant improvements. These results collectively demonstrate that the proposed RNB is highly adaptable to datasets of varying sizes and effectively balances discrimination power with generalization capability. Such performance highlights the robustness and versatility of RNB in diverse classification tasks.

3.4.3 Experimental Analysis

In the statistical significance tests shown in Table 3.3, the proposed approach significantly outperforms CAWNB [22], CWANB [42], WANBIA [21], DEAWNB [41], CFW [19], DAWNB [32], TCSFS-NB [18] and GNB[165] on 8, 9, 10, 12, 14, 17, 17 and 23 datasets, respectively. Compared with the original NB, on more than half of the datasets, the proposed RNB achieves statistically significant improvements. Compared with the previous best algorithm, CAWNB [22], the proposed RNB achieves statistically significant improvements on 8 datasets, which demonstrates the effectiveness of the proposed approach.

Table 3.4 summarizes the results for statistical significance tests. For each entry $u(v)$, u is the number of datasets on which the proposed RNB outperforms the corresponding competitor, and v is the number of datasets on which the performance gain is statistically significant with significance level $p = 0.05$. Table 3.4 shows that on average the classification accuracies on more than two-thirds of 33 datasets improve and half of them are statistically significant. It hence can be concluded that the proposed RNB outperforms all compared approaches.

Table 3.4: Summary of the results for statistical significance tests. For example, RNB outperforms CAWNB on 21 datasets, among which 8 are statistically significant.

Algorithm	GNB [165]	TCSFS-NB [18]	CFW [19]	CWANB [42]	CAWNB [22]	NB [175]	DAWNB [32]	DEAWNB [41]	WANBIA [21]
RNB	29(23)	28(17)	24(14)	24(9)	21(8)	25(18)	26(17)	26(12)	22(10)

From the experimental results, it can be seen that the proposed regularized naive Bayes achieves a remarkable performance improvement. The hyper-parameter α is optimized along with class-dependent attribute weights and class-independent attribute weights. The optimal value of α on each dataset is shown in Table 3.5, together with the number of instances and the number of instances per class. The values of α^* vary on different datasets. In general, the larger the dataset, the higher the α^* value.

To better see the trend, the average value of α^* across datasets and the performance gain of the proposed RNB against the second best algorithm, CAWNB [22], are summarized in Table 3.6. The 33 datasets are divided into small and large datasets according to the number of instances per class, e.g. if it is larger than 500, the dataset is considered

Table 3.5: The number of instances, instances per class and the optimal value of α on 33 datasets.

Datasets	Instance	Instance/class	α^*
anneal	898	150	1.0000
audiology	226	9	0.9875
balance-scale	625	208	0.7293
breast-cancer	286	143	0.4591
breast-w	699	350	0.4312
colic	368	184	0.4991
credit-a	690	345	0.4486
credit-g	1000	500	0.5258
diabetes	768	384	0.2839
glass	214	31	0.9741
heart-c	303	61	0.3981
hear-h	294	59	0.7923
heat-statlog	270	135	0.5767
hepetitis	155	78	0.5815
hypothroid	3772	943	1.0000
ionophere	351	176	0.4925
iris	150	50	0.1935
kr-vs-kp	3196	1598	1.0000
labor	57	29	1.0000
letter	20000	769	0.5536
lymphoraphy	148	37	0.8762
mushroom	8124	4062	1.0000
primary-tumor	339	16	1.0000
segment	2310	330	0.0223
sick	3772	1886	0.9155
sonar	208	104	0.0000
soybean	683	36	0.0000
splice	3190	1063	0.1932
vehicle	846	212	0.4588
vote	435	218	0.0000
vowel	990	90	0.4530
wave-5000	5000	1667	0.8445
zoo	101	14	0.9916

large, and small otherwise. Table 3.6 shows that for small datasets, the average α^* value is significantly smaller than that for large datasets. This indicates that α^* could be automatically adjusted during optimization so that for small datasets, α^* will be small to favor the generalization capability, whereas for large datasets, α^* will be large to favor the discrimination power. It can also be seen that the proposed RNB indeed demonstrates good generalization capabilities for small datasets by achieving a larger performance gain than that on large datasets.

Table 3.6: The average value of α^* and the performance gain of the proposed RNB against CAWNB [22] for small/large datasets.

	Small datasets	Large datasets
Average α^*	0.5460	0.7541
Performance gain(%)	1.3267	0.2978

3.5 Summary

In this paper, after a thorough literature review of the state-of-the-art attribute-weighting naive Bayes methods, we find that class-dependent attribute-weighting naive Bayes has poor generalization capabilities on relatively small datasets. Therefore, we propose to add a regularization term to alleviate the problem. The regularization term is extracted from a simpler naive Bayes which has better generalization capabilities. The proposed regularized naive Bayes is hence derived by integrating the regularization term into the CAWNB. A gradient-descent-based optimization procedure has been designed to derive the optimal model parameters including class-dependent weight matrix $\boldsymbol{W}$, class-independent weight vector $\boldsymbol{w}$ and the hyper-parameter α. Experimental results on the 33 datasets validate the effectiveness of the proposed RNB. The proposed method outperforms the previous best algorithm CAWNB on 21 datasets, of which 8 are statistically significant, and the average performance gain on the 33 datasets is more than 1%.

Chapter 4

A Semi-Supervised Adaptive Discriminative Discretization Method Improving Discrimination Power of Regularized Naive Bayes

Recently, many improved naive Bayes methods have been developed with enhanced discrimination capabilities[1]. Among them, regularized naive Bayes (RNB) produces excellent performance by balancing the discrimination power and generalization capability. Data discretization is important in naive Bayes. By grouping similar values into one interval, the data distribution could be better estimated. However, existing methods including RNB often discretize the data into too few intervals, which may result in a significant information loss. To address this problem, we propose a semi-supervised adaptive discriminative discretization framework for naive Bayes, which could better estimate the data distribution by utilizing both labeled data and unlabeled data through pseudo-labeling techniques. The proposed method also significantly reduces the information loss during discretization by utilizing an adaptive discriminative discretization

[1] This work has been published in Expert System with Applications [177]

scheme, and hence greatly improves the discrimination power of classifiers. The proposed RNB+, i.e., regularized naive Bayes utilizing the proposed discretization framework, is systematically evaluated on a wide range of machine-learning datasets. It significantly and consistently outperforms state-of-the-art NB classifiers.

4.1 Introduction

Naive Bayes (NB) has been widely used in many machine-learning tasks because of its simplicity and efficiency [20, 178–184]. It could well handle different data types such as numerical and categorical ones. Naive Bayes assumes that features are independent conditioned on the classification variable, while such an assumption often does not hold [9, 160], which may degrade the classification performance. Numerous improved NB classifiers have been developed to alleviate this problem, which can be broadly divided into five categories: structure extension [11, 12, 182], instance selection [13, 90], instance weighting [20, 43], attribute selection [17, 18, 60, 185] and attribute weighting [5, 9, 19, 21, 22, 186].

Among these, attribute weighting NB classifiers comparably perform better [5, 9, 19, 21, 22, 32]. In WANBIA, attributes are weighted differently according to the feature importance using the classification feedback [21]. In class-specific attribute weighted naive Bayes (CAWNB), different weights are assigned to the attributes of different classes to enhance the discrimination power [22]. Most recently, regularized naive Bayes has been developed to improve the classification performance by balancing the generalization capability and discrimination power of the classifier automatically through a gradient descent optimization using the classification feedback [9]. These methods enhance the discrimination power of NB classifiers by feature weighting, but overlook the issues on data discretization.

The goal of data discretization is to find a set of cut points to optimally discretize numerical attributes, reducing the inconsistency rate while preserving the discriminant information [81]. However, improving generalization ability by reducing the inconsistency rate and maintaining the discrimination power are two opposite goals. On the one hand, by grouping similar values into one interval, more samples can be used to better estimate the distribution of the interval, leading to better generalization abilities, but at the cost of losing discriminant information. On the other hand, without data discretization, the discriminative ability is retained to the greatest extent, but the generalization ability is poor. A well-designed data discretization method should balance the trade-off between preserving discriminative ability and improving generalization ability.

In literature, many discretizers follow this design principle [23, 25, 44]. In CAIM, a greedy algorithm is utilized to approximately find the global optimum by simultaneously minimizing the number of intervals and maximizing the class-attribute interdependence, but CAIM does not guarantee to find the global optimum [23]. In MDLP, an entropy-based discretization criterion is utilized to select the cut points by maximizing the entropy of the data, which splits the attribute into intervals in a top-down manner [44]. To avoid excessive splitting, a stopping criterion is defined. But this stopping criterion often leads to an early stop in the splitting process, very few discretization intervals and hence a significant information loss. Despite all these problems, it is surprising that MDLP is often used in advanced naive Bayes classifiers and yields satisfactory performance [9, 20–22].

The aforementioned discretization methods are often known as supervised discretization methods [23, 25, 44], where the class information is utilized to guide the discretization process. In literature, unsupervised methods such as equal-width discretization [23] and equal-frequency discretization [23] are also used, which do not require the class information. The collected data are often unlabeled and labeling data is often expensive because it needs the expert knowledge [187]. Hence, there is often a huge amount of unlabeled data, whereas only a small portion is labeled. In this case, a semi-supervised discretization scheme is preferred to utilize both labeled and unlabeled data.

In this paper, we propose a semi-supervised adaptive discriminative discretization (SADD) to address the problem of previous methods, targeting at balancing the discrimination power and generalization ability of NB classifiers. In recent years, semi-supervised methods have been successfully applied in machine learning tasks to improve the generalization ability of models on unseen data and avoid the overfitting problem [187–191]. In the proposed semi-supervised discretization method, unlabeled data is first assigned a pseudo label by using a simple classification model such as k-Nearest Neighbors (K-NN) classifier [187, 189]. Then, the pseudo-labeled data is integrated with the labeled data to provide more discriminant information for the discretization method. With the help of pseudo-labeled data, the intrinsic data structure could be better discovered in which the discriminative ability and generalization ability of subsequently trained classifiers can be greatly enhanced.

After pseudo-labeling, an adaptive discriminative discretization scheme is proposed in this paper. The proposed semi-supervised framework could better discover the intrinsic data properties, so that the data distribution could be better estimated. Data is often discretized to improve the generalization ability by grouping similar values into one interval, but too few intervals will result in a significant information loss and too few samples in the interval will result in poor generalization performance. The proposed SADD explicitly addresses the problem of early stop in MDLP [44] by using an adaptive discriminative discretization scheme, and hence resolves the issue of significant discriminant information loss in MDLP. As a result, each interval has a sufficient number of samples to reliably estimate the likelihood probabilities in naive Bayes so that the naive Bayes can generalize well on unseen data, and a sufficient number of intervals are retained to maintain the discriminant information. In other words, the proposed SADD well balances the discrimination power and generalization ability of the data discretizer.

The proposed SADD is integrated with the recent development of NB classifier, regularized naive Bayes (RNB) [9], and the integrated method is named RNB+. Compared to RNB, the proposed RNB+ well addresses the early stopping problem in data discretization of RNB and preserves the discriminant information of the data, and hence better balances the discrimination power and generalization ability. The proposed methods are evaluated on a wide range of machine-learning datasets for various applications. Experimental results show that the proposed SADD significantly outperforms the widely used discretization methods [23, 25, 44] and the proposed RNB+ significantly outperforms the state-of-the-art NB classifiers [9, 20–22].

4.2 Related Work

In naive Bayes classifiers and many other classifiers [5, 9, 19–22, 32], numerical attributes are often discretized to group similar values into one bin, to address the problem that numerical attributes often have lots of noisy samples resulting in poor generalization capabilities. Existing discretization methods can be broadly divided into unsupervised, semi-supervised and supervised methods, depending on whether the class information is used [76]. Unsupervised discretization methods include equal-frequency, equal-width discretization [23], proportional k-interval discretization (PKID) [84] and fixed frequency discretization (FFD) [115]. Equal-frequency discretization divides the attribute set into intervals with the same number of instances, while equal-width discretization divides the attribute set into intervals with the same length [23]. PKID adjusts the number and size of intervals proportional to the number of training instances [84]. FFD divides the data into intervals with a pre-defined frequency [115] Supervised discretization methods include MDLP [44], other information-based algorithms [23, 192], and statistical algorithms like ChiMerge [88], class-attribute interdependence maximization (CAIM) [23] and class-attribute contingency coefficients (CACC) [25]. Semi-supervised discretization methods are comparatively less studied. [127] developed a semi-supervised framework based on MODL to exploit a mixture of labeled and unlabeled data. In this paper, we mainly review CAIM [23] and MDLP [44] in detail as they follow closely to the design concept of balancing the discrimination power and generalization ability. In CAIM, the boundary point is selected with the maximal interdependence in a top-down manner by using a quanta matrix [23], but the number of generated intervals is too close to the number of classes. To address this problem, [25] developed a discretization method based on class-attribute contingency coefficients to prevent information loss.

Many state-of-the-art NB classifiers such as WANBIA [21], CAWNB [22], AIWNB [20] and RNB [9] utilize the MDLP criterion [44] to discretize numerical attributes in a top-down manner. In MDLP, for each interval, a cut point with the maximum entropy amongst all candidates is selected to split the interval into two, towards the goal of retaining the maximum amount of discriminant information [44]. To avoid the poor generalization ability caused by excessive splitting, a stop criterion is designed based on the concept of information encoding in a communication channel [44]. MDLP often leads to a good classification performance as it balances the discrimination power and generalization ability. However, as shown later in the next section, the early stopping problem during splitting in MDLP often leads to a huge information loss and hence degrades the performance of subsequent classifiers.

4.3 Proposed Method

4.3.1 Problem Analysis of MDLP

Data discretization is crucial to classification performance. As an entropy-based discretization method with strong theoretical background [67, 81, 133, 134, 193, 194], MDLP [44] has been widely used in many state-of-the-art attribute weighting NB classifiers [9, 20–22]. It splits the dynamic range in a top-down manner, i.e., for each attribute, a cut point retaining the maximum amount of discriminant information is selected to divide the current set into two. The discriminant information is measured by the information gain of a cut point d for a given attribute, dividing the current example set $\mathcal{S}$ into two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$. The information gain $G(\mathcal{S}, d)$ is defined as,

$$G(\mathcal{S}, d) = E(\mathcal{S}) - \frac{|\mathcal{S}_1|}{|\mathcal{S}|}E(\mathcal{S}_1) - \frac{|\mathcal{S}_2|}{|\mathcal{S}|}E(\mathcal{S}_2), \tag{4.1}$$

where $E(\mathcal{S})$ is the class entropy as defined below,

$$E(\mathcal{S}) = -\sum_{i=1}^{k} P(C_i, \mathcal{S})log(P(C_i, \mathcal{S})). \tag{4.2}$$

$P(C_i, \mathcal{S})$ is the prior probability of class C_i in $\mathcal{S}$, and k is the number of classes. The binary split in MDLP is applied recursively if Eqn. (4.3) holds, and stops otherwise. Intuitively, the stop criterion will prevent excessively splitting the attribute into too many small intervals with too few samples so that the likelihood probabilities can not be reliably estimated.

$$G(\mathcal{S}, d) > \theta, \tag{4.3}$$

where θ is the adaptive threshold for discretization,

$$\theta = \frac{log_2(N-1)}{N} + \frac{\Delta(\mathcal{S})}{N}, \tag{4.4}$$

where N is the number of samples in $\mathcal{S}$. $\Delta(\mathcal{S})$ is defined as,

$$\Delta(\mathcal{S}) = log_2(3^k - 2) - [kE(\mathcal{S}) - k_1 E(\mathcal{S}_1) - k_2 E(\mathcal{S}_2)], \tag{4.5}$$

where k_1 and k_2 are the number of classes in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Empirical study shows that the threshold θ is often too large compared to the information gain $G(\mathcal{S}, d)$ so that the top-down splitting often stops at an early stage. As a result, a huge amount of discriminant information is lost.

To explore the root cause of this early stop, we take a close examination of the adaptive threshold θ defined in Eqn. (4.4). Empirical study shows that the first term $\frac{log_2(N-1)}{N}$ dominates the adaptive threshold θ, while the second term $\frac{\Delta(\mathcal{S})}{N}$ is a relatively small positive value. We further analyze $\frac{log_2(N-1)}{N}$ by plotting it against N as shown in Fig. 4.2. Apparently when N is large, $\frac{log_2(N-1)}{N}$ is relatively small and the data could be easily split into intervals in a top-down manner. As the split continues, the number of samples N in the interval becomes smaller, leading to a large $\frac{log_2(N-1)}{N}$, and hence it is more difficult to split. This decision criterion follows the design principle of balancing the discriminative ability and generalization ability, and works well when N is large. However, for small N, $\frac{log_2(N-1)}{N}$ is relatively large and hence many attributes with a small number of samples may not split at all at the very beginning. In this case, the attribute is discretized into one bin only, and the discriminant information residing in the attribute is totally lost.

4.3.2 Overview of Proposed SADD Framework for Regularized Naive Bayes

As shown in the previous subsection, the early stopping problem in MDLP may result in a significant loss of discriminant information during discretization. In a broader sense, data discretization helps to improve the generalization ability of naive Bayes classifiers by grouping similar values into one bin, whereas excessive grouping (such as an early stop in MDLP) will result in a significant information loss. It is hence crucial for a data discretizer to balance the generalization ability and discrimination power. Furthermore, supervised discretization methods such as MDLP [44] and CAIM [23] are not capable of handling unlabeled data without any adaptation, and hence the information residing in unlabeled data can't be fully exploited by those supervised discretization methods. Thus, a semi-supervised discretization method exploiting both labeled and unlabeled data is needed.

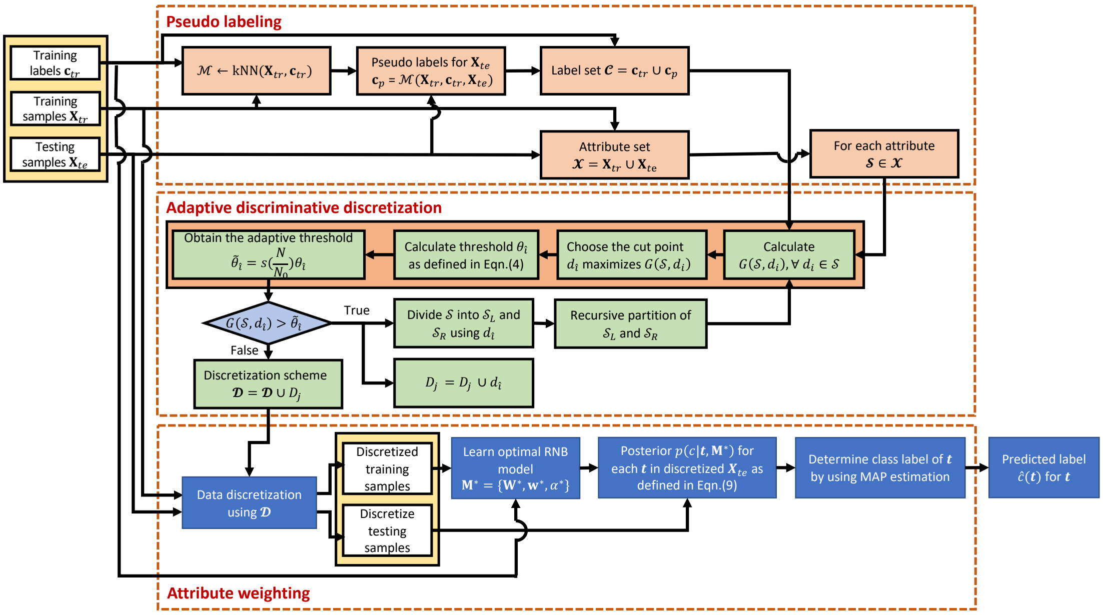

Pseudo labeling
Training labels $\mathbf{c}_{tr}$
Training samples $\mathbf{X}_{tr}$
Testing samples $\mathbf{X}_{te}$
$\mathcal{M} \leftarrow \text{kNN}(\mathbf{X}_{tr}, \mathbf{c}_{tr})$
Pseudo labels for $\mathbf{X}_{te}$ $\mathbf{c}_p = \mathcal{M}(\mathbf{X}_{tr}, \mathbf{c}_{tr}, \mathbf{X}_{te})$
Label set $\mathcal{C} = \mathbf{c}_{tr} \cup \mathbf{c}_p$
Attribute set $\mathcal{X} = \mathbf{X}_{tr} \cup \mathbf{X}_{te}$
For each attribute $\mathcal{S} \in \mathcal{X}$
Adaptive discriminative discretization
Obtain the adaptive threshold $\tilde{\theta}_{\hat{\imath}} = s(\frac{N}{N_0})\theta_{\hat{\imath}}$
Calculate threshold $\theta_{\hat{\imath}}$ as defined in Eqn.(4)
Choose the cut point $d_{\hat{\imath}}$ maximizes $G(\mathcal{S}, d_i)$
Calculate $G(\mathcal{S}, d_i), \forall\, d_i \in \mathcal{S}$
$G(\mathcal{S}, d_{\hat{\imath}}) > \tilde{\theta}_{\hat{\imath}}$
True
False
Divide $\mathcal{S}$ into $\mathcal{S}_L$ and $\mathcal{S}_R$ using $d_{\hat{\imath}}$
Recursive partition of $\mathcal{S}_L$ and $\mathcal{S}_R$
Discretization scheme $\mathcal{D} = \mathcal{D} \cup D_j$
$D_j = D_j \cup d_{\hat{\imath}}$
Data discretization using $\mathcal{D}$
Discretized training samples
Discretize testing samples
Learn optimal RNB model $\mathbf{M}^* = \{\mathbf{W}^*, \mathbf{w}^*, \alpha^*\}$
Posterior $p(c|\boldsymbol{t}, \mathbf{M}^*)$ for each $\boldsymbol{t}$ in discretized $\boldsymbol{X}_{te}$ as defined in Eqn.(9)
Determine class label of $\boldsymbol{t}$ by using MAP estimation
Predicted label $\hat{c}(\boldsymbol{t})$ for $\boldsymbol{t}$
Attribute weighting

Figure 4.1: The proposed SADD for regularized naive Bayes including pseudo labeling, adaptive discriminative discretization and attribute weighting processes.

To address these problems, we propose a Semi-supervised Adaptive Discriminative Discretization (SADD) method for regularized naive Bayes (The integrated method is also known as RNB+), as shown in Fig. 6.1. 1) First of all, a semi-supervised technique is designed to generate the pseudo labels for the unlabeled data, so that the intrinsic data properties in both labeled and unlabeled data could be exploited to better estimate the data statistics. In this paper, the k-NN classifier is applied to generate the pseudo labels. 2) Secondly, an adaptive discriminative discretization scheme is designed to discretize the attribute set, where a new adaptive thresholding strategy is designed to balance the number of intervals required to retain the sufficient discrimination power and the number of samples in an interval to retain the sufficient generalization ability. 3) We consider the trade-off between the generalization ability and discrimination power not only in data discretization, but also in classifier design such as feature weighting. Following this design principle, the proposed SADD is integrated with regularized naive Bayes, in which the attributes are weighted automatically balancing the discrimination power and generalization ability of the classifier.

4.3.3 Proposed Semi-supervised Adaptive Discriminative Discretization

4.3.3.1 Pseudo-labeling

Semi-supervised techniques have proven to be a powerful paradigm for utilizing unlabeled data to improve the generalization ability of learning models relying solely on labeled data [188–191]. Among various semi-supervised methods, pseudo-labeling techniques are effective to tackle the problem, which can be easily integrated with traditional supervised classification algorithms [187]. More specifically, let $\boldsymbol{X}_l$ be the labeled data with class variables $\boldsymbol{c}_l$ and $\boldsymbol{X}_u$ be the unlabeled data, the pseudo labels $\boldsymbol{c}_p$ for $\boldsymbol{X}_u$ can be derived by,

$$\boldsymbol{c}_p = \mathcal{M}(\boldsymbol{X}_l, \boldsymbol{c}_l, \boldsymbol{X}_u), \tag{4.6}$$

where $\mathcal{M}$ represents a pseudo-labeling algorithm. There are two main approaches to generate the pseudo labels: single-classifier and multi-classifier methods. In a single-classifier model, the pseudo label for each unlabeled instance is derived by using only one classification model. In contrast, the multi-classifier model utilizes the majority voting rule to decide the pseudo label by using several classifiers. To keep the simplicity and effectiveness of the proposed framework, the k-nearest neighbors (k-NN) classifier is applied to generate the pseudo labels for unlabeled data. Then, the pseudo-labeled data and labeled data are combined to better discover the intrinsic data properties and better estimate the data statistics.

4.3.3.2 Adaptive Discriminative Discretization

To address the early stopping problem in previous discretization methods [44], we propose an adaptive discriminative discretization. More specifically, we aim to lower the adaptive threshold used in Eqn. (4.3), especially for small datasets with relatively small N, in order to prevent the early stop and the significant loss of discriminant information during discretization. On the other hand, the new threshold $\tilde{\theta}$ can not be too small. If $\tilde{\theta}$ is approaching zero, each distinct value will become a separate interval, which results in no information loss, but may lead to a poor estimation of data distribution due to insufficient samples in each small interval. Bearing all these in mind, we propose the new threshold $\tilde{\theta}$ as defined below:

$$\tilde{\theta} = s\left(\frac{N}{N_0}\right)\theta, \tag{4.7}$$

where $s(x) = 1/(1+e^{-x})$ is the sigmoid function and N_0 is a constant. As $\frac{N}{N_0}$ is non-negative, it is easy to show that $s(\frac{N}{N_0}) \in (0.5, 1)$. N_0 is used to judge whether there are sufficient samples in the interval. If $N \gg N_0$, i.e., there are sufficient samples, $s(\frac{N}{N_0}) \approx 1$. In this case, although $s(\frac{N}{N_0})$ is relatively large, θ is relatively small, and $\tilde{\theta}$ is small enough so that the top-down split could continue, and the resulting intervals will have sufficient samples to reliably estimate the likelihood probabilities. If $N \approx N_0$, $s(\frac{N}{N_0}) \approx 0.73$, i.e., a significantly lower threshold $\tilde{\theta}$ will be used in the top-down discretization compared to the threshold θ defined in Eqn. (4.4). Consequently, it will encourage further splitting of the interval and hence retain more discriminant information. To prevent excessive splitting, the proposed method has a safeguard mechanism. More specifically, when $N \ll N_0$, $s(\frac{N}{N_0}) \approx 0.5$, i.e., the maximum reduction of the threshold $\tilde{\theta}$ from θ is 50%.

To further illustrate the effect of the adaptive threshold in the proposed method, we plot the value of the adaptive version of $\frac{log_2(N-1)}{N}$ after multiplying it with $s(\frac{N}{N_0})$ for different N_0, as shown in Fig. 4.2. With the small number of samples in an interval, the threshold is greatly decreased from about 0.4 to 0.2 to encourage further splitting, especially for small datasets. When N is large, the new adaptive threshold is smaller than the original one, but very close to it, to encourage the split. In addition, the proposed method is insensitive to the choice of N_0. As shown in Fig. 4.2, the new thresholds for $N_0 = 100$ and $N_0 = 2000$ are quite close when N is small. For large N, the difference does not matter so much as the set will be split into smaller ones anyway. In summary, the proposed SADD could effectively prevent the early stop during data discretization, as

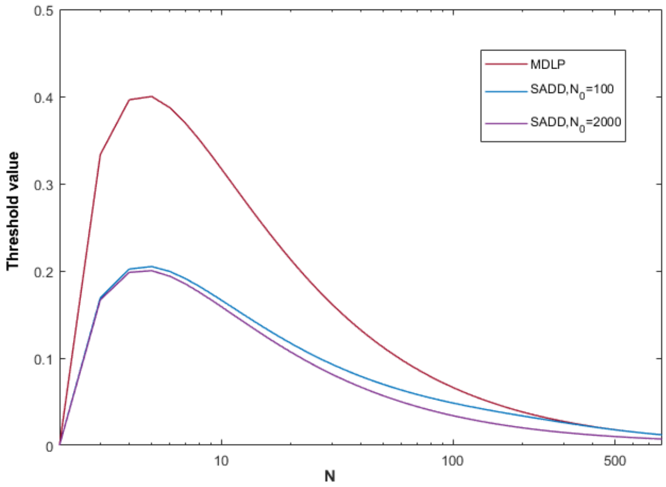


Figure 4.2: Plot of $\frac{log_2(N-1)}{N}$ and its adaptive version $s(\frac{N}{N_0})\frac{log_2(N-1)}{N}$ with different values of N_0.

well as the excessive split. The proposed SADD algorithm is summarized in Algorithm 4.

Algorithm 3 The proposed SADD algorithm

```
Input: Training data $\boldsymbol{X}_l$ with class $\boldsymbol{c}_l$ and testing data $\boldsymbol{X}_u$
Output: Discretization scheme $\boldsymbol{\mathcal{D}} \leftarrow \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m\}$ for $m$-dimensional features
$\boldsymbol{c}_p \leftarrow \mathcal{M}(\boldsymbol{X}_l, \boldsymbol{c}_l, \boldsymbol{X}_u)$        ▷ Derive pseudo labels using Eqn. (4.6)
$\boldsymbol{\mathcal{X}} \leftarrow \boldsymbol{X}_l \cup \boldsymbol{X}_u$        ▷ $\boldsymbol{\mathcal{X}}$ is the set of samples
$\boldsymbol{\mathcal{C}} \leftarrow \boldsymbol{c}_l \cup \boldsymbol{c}_p$        ▷ $\boldsymbol{\mathcal{C}}$ is the set of labels
$\boldsymbol{\mathcal{D}} \leftarrow \emptyset$        ▷ Initialize $\boldsymbol{\mathcal{D}}$ as an empty set
for each $\mathcal{X}_j \in \boldsymbol{\mathcal{X}}$ do
    $\mathcal{D}_j \leftarrow \emptyset$        ▷ $\mathcal{D}_j$ is the discretization scheme for $\mathcal{X}_j$
    $\mathcal{S} \leftarrow \mathcal{X}_j$        ▷ $\mathcal{S}$ is the set of samples to be discretized
    procedure PARTITION($\mathcal{S}, \boldsymbol{\mathcal{C}}$)        ▷ Procedure to partition $\mathcal{S}$ using $\boldsymbol{\mathcal{C}}$
        if $|\mathcal{S}|$ ==1 then        ▷ If there is only one sample in $\mathcal{S}$, return
            return
        end if
        Calculate $G(\mathcal{S}, d_i), \forall d_i \in \mathcal{S}$, as defined in Eqn. (4.1).
        Choose the cut point $d_{\hat{i}}$, $\hat{i} = \arg\max_i G(\mathcal{S}, d_i), \forall d_i \in \mathcal{S}$.
        Calculate the threshold $\theta_{\hat{i}}$ using Eqn. (4.4) for the cut point $d_{\hat{i}}$.
        Calculate the new adaptive threshold $\tilde{\theta}_{\hat{i}} = s\left(\frac{N}{N_0}\right)\theta_{\hat{i}}$.
        if $G(\mathcal{S}, d_{\hat{i}}) > \tilde{\theta}_{\hat{i}}$ then        ▷ The SADD stop criterion
            $\mathcal{D}_j \leftarrow \mathcal{D}_j \cup d_{\hat{i}}$        ▷ Insert $d_{\hat{i}}$ into discretization scheme $\mathcal{D}_j$
            $\mathcal{S}_L \leftarrow \mathcal{S} < d_{\hat{i}}$        ▷ Divide $\mathcal{S}$ into the set smaller than $d_{\hat{i}}$
            $\mathcal{S}_R \leftarrow \mathcal{S} \geq d_{\hat{i}}$        ▷ Divide $\mathcal{S}$ into the set not smaller than $d_{\hat{i}}$
            PARTITION($\mathcal{S}_L, \boldsymbol{\mathcal{C}}$)        ▷ Recursively partition $\mathcal{S}_L$ using $\boldsymbol{\mathcal{C}}$
            PARTITION($\mathcal{S}_R, \boldsymbol{\mathcal{C}}$)        ▷ Recursively partition $\mathcal{S}_R$ using $\boldsymbol{\mathcal{C}}$
        end if
    end procedure
    $\boldsymbol{\mathcal{D}} \leftarrow \boldsymbol{\mathcal{D}} \cup \mathcal{D}_j$        ▷ Add the discretization scheme $\mathcal{D}_j$ into $\boldsymbol{\mathcal{D}}$
end for
return $\boldsymbol{\mathcal{D}}$
```

In the first step of Algorithm 4, pseudo labeling, the k-NN classification model $\mathcal{M}$ is generated by using the labeled training data and used to derive the pseudo labels $\boldsymbol{c}_p$ for unlabeled testing data using Eqn. (4.6). Then, the attribute set $\boldsymbol{\mathcal{X}}$ and label set $\boldsymbol{\mathcal{C}}$ are derived by combining the labeled training data with pseudo-labeled testing data. Then the attributes $\mathcal{X}_j \in \boldsymbol{\mathcal{X}}$ are discretized one at a time. The procedure PARTITION is used to find the optimal cut point $d_{\hat{i}}$ to divide the current sample set $\mathcal{S}$ into two sets $\mathcal{S}_L$ and $\mathcal{S}_R$, where $\hat{i} = \arg\max_i G(\mathcal{S}, d_i), \forall d_i \in \mathcal{S}$ and $G(\mathcal{S}, d_i)$ is the information gain defined in Eqn. (4.1). Then, the PARTITION procedure is recursively applied on $\mathcal{S}_L$ and $\mathcal{S}_R$ to find the optimal cut point to further discretize the attribute. The recursive partition continues as long as the following condition holds:

$$G(\mathcal{S}, d_{\hat{i}}) > \tilde{\theta}_{\hat{i}}, \tag{4.8}$$

where $\tilde{\theta}_{\hat{i}} = s\left(\frac{N}{N_0}\right)\theta_{\hat{i}}$ is the newly defined adaptive threshold, $\theta_{\hat{i}}$ is the threshold defined in Eqn. (4.4) for the cut point $d_{\hat{i}}$ and $s(\cdot)$ is the sigmoid function. The discretization scheme $\mathcal{D}_j$ for attribute $\mathcal{X}_j$ is updated as,

$$\mathcal{D}_j \leftarrow \mathcal{D}_j \cup d_{\hat{i}}. \tag{4.9}$$

For each attribute, the proposed SADD utilizes a greedy hierarchical splitting algorithm to generate a tree-like discretization scheme, as summarized in Algo. 4. It can be shown that the time complexity is $O(n \log n)$ for each attribute, where n is the number of samples. The total time complexity for m attributes is hence $O(mn \log n)$.

4.3.4 Discussion and Analysis

As discussed early, the proposed SADD could effectively prevent the information loss of MDLP. To further analyze this, a case study is presented in Table 4.1, which shows the number of intervals (Num.) and the mutual information (MI) on the "Vowel" dataset after discretization by MDLP [44] and the proposed SADD, respectively. The dataset contains 10 numerical attributes and 11 classes. Most numerical attributes are discretized into 3-4 intervals by MDLP, which can not effectively differentiate 11 classes. After applying the proposed SADD, more intervals could be obtained and hence less

Table 4.1: The comparisons of the number of intervals (Num.) and the mutual information (MI), after data discretization by the proposed SADD and MDLP [44] on "Vowel" dataset.

	Proposed SADD		MDLP	
	MI	Num.	MI	Num.
A_1	1.0921	18	0.9596	8
A_2	1.2180	19	1.0875	9
A_3	0.2998	8	0.1198	3
A_4	0.4488	7	0.3545	4
A_5	0.5909	14	0.4104	4
A_6	0.4347	11	0.2759	4
A_7	0.3287	7	0.2146	3
A_8	0.2447	7	0.1685	3
A_9	0.3009	7	0.1796	3
A_{10}	0.0527	2	0	1
AVG	0.5011	10	0.3770	4.2

discriminant information is lost, as shown in Table 4.1. The proposed SADD can effectively prevent information loss and generate a discretization scheme with a better trade-off between the number of intervals and the number of samples in the intervals. On the one hand, more intervals will retain more discriminant information, but lead to a poor generalization ability as there are too few samples in an interval to reliably estimate the data distribution. On the other hand, too few intervals may result in a huge discriminant information loss, as shown in the early stopping case of MDLP.

4.3.5 Proposed RNB+

MDLP has been widely used in many state-of-the-art naive Bayes classifiers, e.g., AIWNB [20], WANBIA [21], CAWNB [22] and RNB [9]. As shown previously, MDLP may result in a huge information loss and hence the SADD is proposed to address this problem. In this section, we describe how to integrate the proposed SADD with RNB to boost the classification performance of NB classifiers. The integrated method is named as RNB+. We first discretize the data using the proposed SADD so that the data distribution could be better estimated, and then use RNB as the classifier.

We have shown that the proposed SADD could well balance the generalization ability and discrimination power during data discretization. Now we show how the proposed

RNB+ achieves a better trade-off during attribute weighting. To alleviate the conditional independence assumption of NB classifiers, attribute weighting techniques have been widely used in NB classifiers and achieved remarkable performance [9, 20–22]. In WANBIA, the same weight is assigned to the attributes in different classes [21], while in CAWNB, a class-specific weight is assigned to each attribute to capture more data characteristics [22]. But the model complexity increases with more attribute weights, and CAWNB hence may overfit to the data, especially for small datasets. To alleviate this problem, regularized naive Bayes has been recently developed, which regularizes CAWNB by adding a simpler model, i.e., WANBIA, to penalize the model complexity [9].

More specifically, in RNB, the target is to find the optimal model parameters $\boldsymbol{M}$ $=\{\boldsymbol{W}, \boldsymbol{w}, \alpha\}$ to minimize the difference between the posterior derived from the ground-truth label and the estimated posterior for a given instance $\boldsymbol{x}$,

$$P(c|\boldsymbol{x}, \boldsymbol{M}) = \alpha P_D(c|\boldsymbol{x}, \boldsymbol{W}) + (1-\alpha) P_I(c|\boldsymbol{x}, \boldsymbol{w}), \tag{4.10}$$

where $P_D(c|\boldsymbol{x}, \boldsymbol{W})$ is the posterior where attributes are weighted on a class-specific basis and $\boldsymbol{W}$ is the weight matrix. $P_I(c|\boldsymbol{x}, \boldsymbol{w})$ is the posterior where attributes are weighted the same for all classes and $\boldsymbol{w}$ is the weight vector. $P_D(c|\boldsymbol{x}, \boldsymbol{W})$ is a more complex model that could provide more discrimination power, whereas $P_I(c|\boldsymbol{x}, \boldsymbol{w})$ is a simpler model that can provide better generalization ability. In RNB, the optimal model parameters $\boldsymbol{M^*}$ are derived through a gradient-descent algorithm, and the discrimination power and generalization ability are automatically balanced by optimizing α [9]. Finally, the predicted label for each test instance $\boldsymbol{t}$ is obtained by using the MAP estimation as follows:

$$\hat{c}(\boldsymbol{t}) = \arg\max_{c \in C} P(c|\boldsymbol{t}, \boldsymbol{M}^*), \tag{4.11}$$

where C is the set of labels for all classes.

The categorical attributes and numerical attributes are often mixed and NB classifiers can generate a probabilistic model on both data types. However, the numerical attributes often have a large number of distinct values so that the likelihood probability estimated from the frequency of instances with a particular value x_i in the j-th attribute given the class c, $P(A_j = x_i|c)$, can be extremely small. The estimation of $P(A_j = x_i|c)$ may not be reliable due to very few training instances. To address this problem, discretization methods have been developed by grouping similar values into one interval and then sufficient training instances can be used to reliably estimate the likelihood probability. However, many discretization methods, e.g., MDLP [44] in the state-of-the-art NB classifiers [9, 20–22], can't generate a proper discretization scheme and may lead to the huge information loss. Thus, RNB+ is proposed to alleviate this problem and retain the discriminative ability from the discretization perspective. As shown later in the experiments, the proposed RNB+ significantly outperforms the state-of-the-art NB classifiers such as RNB [9], WANBIA [21], CAWNB [22] and AIWNB [20].

4.4 Experimental Results

4.4.1 Experimental Settings

The experiments are divided into two parts under NB classification framework [175]. The proposed SADD is firstly compared with other discretization methods including four supervised discretization, MDLP [44], CAIM [23], CACC [25] and ChiMerge [23], and four unsupervised discretization, Equal-Frequency [23], Equal-Width [23] PKID [84] and FFD [115]. Then, the proposed RNB+ is compared with RNB [9], WANBIA [21], CAWNB [22] and AIWNB [20], which are four recent attribute-weighting NB classifiers. All the competitors are summarized in Table 4.2. The experimental results of PKID [84] and FFD [115] are obtained by using the popular data mining tool, KEEL [195]. The other competitors are implemented by using MATLAB. In the proposed SADD, the k-NN classifier with Euclidean distance is used in pseudo-labeling, where the optimal k is tuned by using the validation set. Specifically, one out of nine folds of the training data is randomly selected as the validation set. The optimal k is derived by using a grid search that produces the highest classification accuracy on the validation set.

The comparison experiments are conducted on a set of machine-learning datasets in various domains such as healthcare, biology, disease diagnosis and business. All the datasets

Table 4.2: Description of competitors: six popular discretization methods and four state-of-the-art NB classifiers.

Discretization methods	
MDLP	Supervised entropy-based top-down discretization
CAIM	Supervised statistical top-down discretization
CACC	Supervised statistical top-down discretization
ChiMerge	Supervised statistical bottom-up discretization
Equal-W	Unsupervised top-down discretization
Equal-F	Unsupervised top-down discretization
PKID	Unsupervised top-down discretization
FFD	Unsupervised top-down discretization
Naive Bayes methods	
WANBIA	Wrapper-based class-independent attribute weighting
CAWNB	Wrapper-based class-specific attribute weighting
AIWNB	Filter-based attribute and instance weighting
RNB	Wrapper-based regularized attribute weighting

are extracted from the UCI machine learning repository [2]. Among them, 12 datasets were used in CACC [25] and the rest of them are selected to enrich the comparison experiments. The number of instances is distributed between 150 and 21048 and the number of attributes is between 4 and 520. The numerical attributes and categorical attributes are mixed in the datasets. Some datasets have missing values, which are replaced by the mean of corresponding numerical attributes or mode of categorical attributes. These 31 benchmark datasets provide a comprehensive evaluation of the proposed SADD and RNB+. The datasets are summarized in Table 6.2.

4.4.2 Comparisons to State-of-the-art Discretization Methods

The proposed SADD is compared with MDLP [44], CAIM [23], CACC [25], ChiMerge [88], Equal-W [23], Equal-F [23], PKID [84] and FFD [115] based on the NB classifier [175]. Table 6.4 summarizes the comparisons to these discretization methods. The classification accuracy of each algorithm on each dataset is derived via stratified 10-fold cross-validation, following the same evaluation protocol used in [9, 20–22]. The average classification accuracies of all algorithms over all the datasets are summarized at the bottom, which can provide a straightforward comparison of different methods. Table 4.5 summarizes that the proposed method significantly outperforms its competitors with a

[2] https://archive.ics.uci.edu/ml/index.php

one-tailed t-test at the significance level of $p = 0.05$. The hyper-parameter N_0 in the proposed SADD is set to 2000 empirically.

As shown in Table 6.4, compared to other discretization methods, the proposed SADD achieves the highest average classification accuracy on most of the datasets. Compared with MDLP [44], CAIM [23], CACC [25] and ChiMerge [88], the proposed SADD obtains the performance gain of 3.11%, 2.80%, 3.00% and 5.49% on average, respectively. Compared with the four unsupervised discretization methods, Equal-W [23], Equal-F [23], PKID [84] and FFD [115], the proposed SADD obtains the average performance gain of

Table 4.3: 31 benchmark datasets are collected from real-world problems in various domains. The number of instances is distributed between 150 and 21048. For the entry $u(v)$ in "Attribute", u denotes the total number of attributes and v denotes the number of categorical attributes.

Dataset	Instance	Attribute	Class	Missing values
Iris	150	4	3	N
Parkinson	195	23	2	N
Seeds	210	7	3	N
Glass	214	10	6	N
Heart	270	13(7)	2	N
Ecoli	336	8	8	N
Bupa	345	6	2	N
Ionophere	351	34(2)	2	N
Movement	360	90	15	N
ILPD	583	10	2	N
Breast	699	9	2	Y
Pima	768	8	2	N
Vowel	990	13	11	N
Biodegradation	1055	41	2	N
Mice Protein	1080	82	8	Y
Yeast	1484	10	8	N
Mfeat-fac	2000	216	10	N
Cardio	2126	23	10	N
Madelon	2600	500	2	N
Spambase	4601	57	2	N
Wave	5000	40	3	N
Wall-Following	5456	24	4	Y
Page-Block	5473	10	5	N
Opdigit	5620	64	10	N
Satellite	6435	36	6	N
Wine	6497	11	7	N
Musk	6598	166	2	N
Anuran	7195	22	4	N
Pendigit	10992	16	10	N
Magic	19020	10	2	N
IndoorLoc	21048	520	3	N

TABLE 4.4: Comparisons between the proposed SADD and other discretization methods. The proposed SADD achieves an average performance gain of 3.11% and 2.80% compared with MDLP and CAIM respectively.

Dataset	SADD	MDLP	CAIM	CACC	ChiMerge	EW	EF	PKID	FFD
Iris	**96.00±4.42**	92.67±6.29	94.00±5.54	93.33±6.67	78.67±9.80	94.67±6.53	92.67±8.14	91.33±9.45	93.33±6.67
Parkinson	**84.02±5.65**	79.46±4.69	81.44±7.29	82.46±6.79	81.02±9.18	79.94±6.41	80.35±7.25	77.26±9.90	77.26±7.84
Seeds	90.95±6.19	87.14±4.29	86.67±5.13	87.62±5.30	80.95±4.26	**91.43±5.13**	88.57±5.71	90.00±8.64	87.62±8.83
Glass	**74.29±4.82**	72.03±8.65	72.92±9.36	65.25±11.14	66.40±10.08	60.75±7.68	68.24±9.01	65.35±8.78	65.71±10.17
Heart	83.70±8.64	83.70±8.64	83.33±8.16	80.74±6.58	83.70±9.40	**84.44±7.73**	82.96±8.31	82.96±4.74	83.70±3.78
Ecoli	**86.62±4.38**	83.10±4.22	81.46±5.58	82.47±5.97	83.38±6.43	85.10±4.26	84.28±5.64	81.58±7.04	82.45±3.05
Bupa	**65.76±10.42**	53.27±9.52	65.24±6.98	63.24±6.30	64.03±8.58	62.61±8.22	58.55±7.53	61.73±7.41	62.87±8.43
Ionophere	**90.62±5.19**	89.52±5.12	88.64±4.11	88.92±4.40	75.83±6.21	90.32±4.41	89.77±5.47	89.16±6.11	89.17±5.09
Movement	**77.77±6.72**	62.10±7.37	71.97±7.94	71.14±7.41	71.41±6.83	70.32±7.17	72.18±6.59	64.87±8.37	67.37±10.45
ILPD	67.05±4.18	64.82±4.45	65.51±4.28	66.03±4.19	65.00±2.68	67.75±2.18	67.40±4.27	67.56±4.89	**68.09±2.91**
Breast	97.42±1.41	97.14±1.43	97.28±1.50	97.28±1.50	95.85±2.67	97.42±1.55	97.42±1.67	**97.43±1.90**	97.43±1.90
Pima	76.82±4.34	73.69±4.70	74.21±5.80	74.21±3.92	72.26±5.17	**77.21±2.80**	74.61±3.61	74.82±4.37	75.35±4.80
Vowel	75.76±4.93	59.09±4.45	64.34±5.27	60.71±6.69	61.11±3.23	67.58±4.87	64.24±5.15	**89.63±1.37**	60.00±6.69
Biodegradation	**81.89±2.80**	80.85±2.76	81.70±2.48	81.70±2.75	78.67±4.12	80.19±2.08	81.04±3.19	80.38±3.22	80.00±3.10
Mice Protein	**98.06±0.77**	93.98±2.60	93.34±2.63	91.67±2.60	92.40±3.55	94.63±2.56	93.05±2.61	93.14±2.05	93.79±2.34
Yeast	**59.79±3.89**	57.15±3.56	57.49±3.54	56.01±2.88	58.23±4.60	58.64±4.80	55.68±4.93	54.31±2.71	53.57±2.81
Mfeat-fac	**94.80±1.95**	93.15±2.15	93.70±2.02	93.70±2.23	93.15±2.08	93.3±2.35	92.70±2.50	92.15±2.08	82.79±2.42
Cardio	81.19±1.41	79.68±1.66	80.34±1.78	79.35±1.38	78.12±1.83	79.25±1.61	77.80±2.10	80.15±1.76	**81.28±2.76**
Madelon	**64.65±3.79**	61.92±3.34	58.69±2.85	50.00±0.00	58.96±3.98	50.00±0.00	50.00±0.00	55.35±2.47	52.92±3.62
Spambase	90.18±1.72	89.63±1.45	89.85±1.53	90.00±1.63	89.42±1.01	85.53±1.97	89.87±1.41	**95.60±0.85**	89.33±1.39
Wave	**80.70±1.00**	80.18±1.00	80.60±1.30	80.32 ± 1.28	78.80±1.04	80.16±0.83	80.24±0.86	79.10±1.60	78.60±1.18
Wall-Following	**90.96±0.96**	89.24±1.29	88.11±1.06	87.81±1.11	71.15±1.60	80.96±1.41	84.26±1.33	60.91±3.62	86.33±1.42
Page-Block	**93.93±1.41**	93.62±1.62	93.17±1.25	93.79±1.22	91.25±1.18	92.54±0.80	88.86±1.57	91.78±1.07	92.98±0.81
Opdigit	**92.65±0.51**	92.38±0.40	92.22±0.64	92.31±0.82	91.76±0.95	92.46±0.74	91.80±0.93	92.19±1.16	92.15±1.15
Satellite	**82.45±1.46**	82.14±1.40	82.02±1.43	82.08±1.40	79.52±1.55	81.18±1.16	81.15±1.31	82.10±1.43	82.14±1.49
Wine	50.42±1.00	49.19±1.42	49.92±0.86	**50.99±1.62**	48.47±1.49	49.42±1.46	47.87±1.07	51.82±1.43	**53.21±1.78**
Musk	**92.98±0.79**	91.76±0.93	85.50±1.54	89.83±0.75	81.60±2.05	84.18±1.85	83.62±1.54	91.13±0.78	61.68±1.61
Anuran	**90.62±1.09**	89.92±1.24	89.42±1.29	89.26±1.41	81.86±1.38	89.33±1.30	89.01±1.00	89.41±1.25	88.27±1.57
Pendigit	**88.43±0.61**	88.10±0.84	87.92±0.72	88.07±0.75	86.96±0.65	87.33±0.83	87.25±0.82	87.24±0.91	86.61±0.79
Magic	**78.13±0.49**	77.67±0.56	75.57±0.57	76.15±0.68	73.26±0.78	74.67±0.51	76.55±0.81	77.78±0.89	77.32±0.82
IndoorLoc	**65.55±0.76**	59.29±1.22	64.18±0.84	64.80±0.80	59.24±0.81	59.94±1.25	41.65±0.86	61.68±0.78	63.19±0.52
AVG	82.07	78.96	79.27	79.07	76.58	78.81	77.86	78.58	78.10

3.26%, 4.21%, 3.49% and 3.97%, respectively. These results demonstrate the superior performance of the proposed SADD.

Table 4.5 summarizes the results for statistical significance tests. The proposed SADD achieves the best performance on most of the datasets, and the performance gains on many of them are statistically significant. Specifically, the proposed SADD outperforms MDLP [44], CAIM [23], CACC [25], ChiMerge [88], Equal-W [23], Equal-F [23], PKID [84] and FFD [115] on 31, 31, 30, 31, 27, 30, 27 and 26 datasets respectively, among which 19, 13, 15, 25, 18, 21, 19 and 18 are statistically significant.

TABLE 4.5: Summary of statistical significance tests on different discretization methods. For each entry $u(v)$, u is the number of datasets on which the proposed SADD outperforms other discretization methods, and v is the number of datasets on which the performance gain is statistically significant with the significance level of $p = 0.05$.

	MDLP	CAIM	CACC	ChiMerge	EW	EF	PKID	FFD
SADD	31(19)	31(13)	30(15)	31(25)	27(19)	30(21)	27(18)	26(18)

4.4.3 Analysis and Discussion of Proposed SADD against MDLP

Table 4.6: The performance gain (PG) of the proposed SADD over MDLP [44] on 31 datasets, and the number of features discretized into the various number of intervals (Num) by the two methods.

	PG(%)	Proposed SADD						MDLP					
Num Dataset		1	2	3	4	5	>5	1	2	3	4	5	>5
Iris	3.33	-	-	1	2	1	-	-	2	2	-	-	-
Parkinson	4.55	1	9	6	4	3	-	2	18	3	-	-	-
Seeds	3.81	-	-	1	2	4	-	-	3	-	3	1	-
Glass	2.25	1	-	1	1	1	6	3	3	2	2	-	-
Heart	0.00	1	5	-	-	-	-	3	3	-	-	-	-
Ecoli	3.52	2	1	1	1	1	2	2	3	3	-	-	-
Bupa	12.49	1	3	2	-	-	-	5	1	-	-	-	-
Ionophere	1.10	-	-	2	3	7	20	1	1	12	2	12	4
Movement	15.67	-	1	5	10	7	67	21	25	29	15	-	-
ILPD	2.23	2	4	2	2	-	-	5	5	-	-	-	-
Breast	0.29	-	1	2	3	1	2	-	1	5	2	-	1
Pima	3.13	-	3	1	4	-	-	2	4	1	1	-	-
Vowel	16.67	-	1	-	-	-	9	1	-	4	3	-	2
Biodegradation	1.04	4	10	14	7	2	4	8	15	15	2	1	-
Mice Protein	4.08	2	7	-	5	13	55	4	25	16	15	9	13
Yeast	2.63	2	3	-	3	2	-	4	2	3	1	-	-
Mfeat-fac	1.65	0	2	5	13	32	164	1	8	59	60	58	30
Cardio	1.51	1	2	2	2	1	15	2	2	5	4	2	8
Madelon	2.73	484	7	5	2	2	-	487	9	1	3	-	-
Spambase	0.54	2	17	17	11	3	7	2	29	15	5	4	2
Wave	0.52	2	-	1	-	1	17	2	-	2	3	2	12
Wall-Following	1.72	-	-	-	-	-	24	-	-	-	-	-	24
Page-block	0.31	-	-	-	-	1	9	-	-	-	-	2	8
Opdigit	0.27	7	7	3	3	8	36	7	9	3	11	18	16
Satellite	0.31	-	-	-	-	-	36	-	-	-	-	-	36
Wine	1.23	1	-	2	1	3	4	1	1	3	4	2	-
Musk	1.23	1	1	3	5	-	156	3	3	8	7	10	135
Anuran	0.69	-	1	-	-	-	21	-	1	-	-	-	21
Pendigit	0.33	-	-	-	-	-	16	-	-	-	-	-	16
Magic	0.46	-	-	-	1	2	7	-	-	-	1	2	7
IndoorLoc	6.25	115	84	53	33	26	39	191	91	37	16	5	2

The proposed SADD is based on the MDLP [44] method but performs significantly better. In Section 4.3.3, we show that in theory the proposed SADD could effectively prevent the information loss and indeed the proposed SADD does significantly outperform MDLP [44] on most datasets as shown in the previous subsection. To analyze the underlying reasons why the proposed SADD performs better than MDLP [44], Table 4.6 summarizes the number of features discretized into the various number of intervals by MDLP [44] and the proposed SADD, respectively, and the performance gain of SADD on each dataset compared with MDLP [44]. For example, for six features of the “Bupa” dataset, five features are discretized into one interval and one is discretized into two intervals by MDLP [44], which leads to a huge information loss. When an attribute is discretized into one interval, the attribute values for all classes are the same. As a result, this attribute can not be used to differentiate different classes and hence the discriminant information residing in the attribute is totally lost. By utilizing the proposed discretization, most features are discretized into more intervals, and hence the discriminant information is better preserved.

As shown in Table 4.6, the proposed SADD performs best on almost all datasets. On datasets with little discriminant information loss during discretization such as "Anuran", "Magic", "Page-Block", "Pendigit" and "Satellite", the performance gains on these datasets are relatively small. On datasets with significant information loss such as "Bupa", "Mice Protein", "Movement", "Parkinson" and "Vowel", the performance gains are high, e.g., the performance gains on "Bupa", "Movement" and "Vowel" are more than 10%. Table 4.6 clearly demonstrates that the proposed SADD could well address the problem of discriminant information loss in MDLP. It generates a proper number of intervals to preserve the discrimination power of classification algorithms, and at the same time retain the generalization capability.

To analyze the performance gains of the proposed SADD over MDLP [44] on different datasets, we group the datasets according to the instance size and the feature size, respectively. 1) In terms of instance size, the proposed SADD greatly enhances the discrimination power of NB classifier on both relatively small datasets (# of Inst. $\leq$ 1000) and relatively large datasets (# of Inst. $>$ 1000), with an average performance gain of 5.31% and 1.53%, respectively. The performance gains on relatively small datasets are more significant because MDLP is more likely to stop the top-down split in the early stages for small datasets. As shown in Fig. 4.2, the large threshold for a small N may cause the early stop of MDLP, and hence lead to a significant performance drop of MDLP, while the proposed SADD well tackles this problem by utilizing a significantly smaller threshold. 2) In terms of feature size, the proposed SADD greatly enhances the discrimination power of NB classifier on both datasets with relatively few features (# of Feat. $\leq$ 50) and datasets with relatively many features (# of Feat. $>$ 50), with an average performance gain of 2.79% and 4.05%, respectively. The performance gains on datasets with more features are more significant because naive Bayes could aggregate the discriminant information gains of more features, resulting in better performance.

4.4.4 Comparisons to State-of-the-art Discretization Methods for Semi-supervised Learning

To evaluate the proposed SADD in a more challenging scenario for semi-supervised learning, we follow the experimental setting in [190, 191], where 40% of training samples are randomly selected as labeled data and the rest are treated as unlabeled data. The proposed SADD utilizes pseudo-labeling techniques to label the unlabeled training data for discretization. The comparisons to four supervised discretization methods under this setting are summarized in Table 4.7. The results of unsupervised discretization methods such as Equal-W [23], Equal-F [23], PKID [84] and FFD [115] may refer back to Table 6.4. As shown in Table 4.7, the proposed SADD achieves an improvement of 3.55% on average compared with MDLP [44]. Compared with CAIM [23], CACC [25] and ChiMerge [88], the proposed SADD obtains the improvements of 1.53%, 2.24% and 4.90%, respectively.

Table 4.8 summarizes the results for statistical significance tests. Among 31 datasets, the proposed SADD outperforms MDLP [44], CAIM [23], CACC [25] and ChiMerge [88] on 31, 29, 30 and 31 datasets respectively, among which 15, 9, 11 and 25 are statistically significant.

4.4.5 Comparisons to State-of-the-art Naive Bayes Classifiers

The proposed SADD can be integrated with not only regularized naïve Bayes, but also other naïve Bayes classifiers. To demonstrate the performance gain brought by the proposed SADD, we integrate it with CAWNB [22], WANBIA [21], AIWNB^E [20] and AIWNB^L [20] and RNB [9] resulting in CAWNB+, WANBIA+, AIWNB^E+, AIWNB^L+ and RNB+ respectively. Note that the MDLP discretization scheme was previously utilized in these NB classifiers. The comparison results are summarized in Table 4.9. As shown in Table 4.9, the proposed SADD has greatly enhanced the performance of these state-of-the-art NB classifiers, and the performance gains on CAWNB, WANBIA, AIWNB^E, AIWNB^L and RNB are 1.99%, 1.82%, 2.71%, 2.16% and 2.16%, respectively. These results demonstrate that the proposed SADD discretization scheme can be

Table 4.7: Comparisons between the proposed SADD and other supervised discretization methods where 40% of training samples are labeled data and the rest are unlabeled data. The proposed SADD obtains an average performance gain of 3.55% compared with MDLP [44].

Dataset	SADD	MDLP	CAIM	CACC	ChiMerge
Iris	**96.00±4.66**	95.33±5.49	92.00±6.89	92.67±5.84	76.67±11.86
Parkinson	**83.07±7.18**	79.88±5.66	81.41±9.12	79.96±7.18	78.60±6.51
Seeds	**89.05±5.04**	88.57±6.02	87.62±6.02	87.62±7.17	81.90±7.03
Glass	**69.62±12.95**	59.92±8.72	68.71±11.80	66.85±10.21	64.37±10.43
Heart	**84.81±8.63**	84.44±9.04	84.44±8.69	82.96±9.43	82.96±9.27
Ecoli	**86.13±6.34**	83.13±6.30	82.80±4.37	83.68±6.18	81.88±6.54
Bupa	**64.03±3.99**	55.63±7.19	62.56±5.86	63.99±6.69	62.01±8.79
Ionophere	**90.35±3.92**	90.09±6.38	89.49±5.75	88.62±4.95	78.06±5.69
Movement	**72.04±7.72**	38.74±7.92	66.90±12.33	70.86±8.55	70.50±6.20
ILPD	**68.44±4.22**	65.34±5.12	67.57±5.80	66.89±5.73	64.31±3.00
Breast	**97.57±2.03**	97.28±1.43	97.14±1.79	97.14±1.79	94.28±2.94
Pima	**77.08±3.95**	74.73±4.20	74.86±5.16	74.87±2.89	71.34±5.82
Vowel	**65.05±4.37**	47.17±8.36	63.94±6.95	52.73±6.96	60.61±4.39
Biodegradation	**81.23±3.43**	77.82±3.85	80.38±3.16	81.23±2.98	77.82±3.58
Mice Protein	**94.62±3.75**	94.53±2.62	93.23±3.02	91.66±3.18	92.12±3.56
Yeast	**59.85±4.28**	58.03±4.29	57.96±3.19	55.81±3.64	57.02±4.97
Mfeat-fac	**93.95±2.28**	92.85±1.90	93.85±2.25	93.60±2.54	93.15±2.17
Cardio	80.10±1.76	78.03±2.35	**80.20±2.66**	79.78±1.62	77.38±1.70
Madelon	**63.31±4.03**	60.42±3.89	58.15±3.88	50.00±0.00	59.31±4.09
Spambase	**90.24±1.58**	90.09±1.58	89.70±1.58	90.02±1.50	89.26±1.10
Wave	**80.24±1.23**	79.76±1.28	79.92±1.08	74.80±0.97	78.72±1.24
Wall-Following	**90.16±1.24**	88.36±1.36	86.82±2.13	88.64±1.80	71.28±1.71
Page-Block	**93.92±1.34**	93.64±1.31	93.09±1.15	93.44±1.57	91.54±1.05
Opdigit	**92.46±0.65**	91.57±0.78	92.40±0.70	92.28±0.67	92.01±0.78
Satellite	**82.44±1.41**	81.79±1.43	81.97±1.43	82.07±1.28	79.60±1.54
Wine	49.38±1.95	48.75±1.02	**50.01±1.86**	49.47±2.03	48.52±1.52
Musk	**91.94±0.77**	90.27±1.00	86.22±2.00	89.80±0.78	78.18±1.94
Anuran	**90.40±1.28**	89.58±1.43	89.28±1.46	89.21±1.57	81.88±1.40
Pendigit	**88.27±0.75**	87.35±0.69	87.67±0.55	88.12±0.69	87.02±0.82
Magic	**76.95±0.47**	76.77±0.77	75.74±0.48	76.04±0.58	73.36±0.92
IndoorLoc	**62.91±1.20**	55.68±0.89	62.24±0.93	61.50±1.12	58.28±1.06
AVG	80.83	77.28	79.30	78.59	75.93

Table 4.8: Summary of statistical significance tests of the proposed SADD over supervised discretization methods when 40% of training samples are labeled data while the rest are unlabeled.

	MDLP	CAIM	CACC	ChiMerge
SADD	31(15)	29(9)	30(11)	31(25)

seamlessly integrated with various naïve Bayes classifiers and significantly improve their performance.

Table 4.10 summarizes the statistical significance tests of the proposed SADD over MDLP on various NB classifiers. By utilizing the proposed SADD discretization scheme, CAWNB+, WANBIA+, AIWNB^E+, AIWNB^L+ and RNB+ achieve the higher classification performance on most of the datasets than their counterparts, CAWNB, WANBIA,

TABLE 4.9: Summary of performance gain brought by the proposed SADD on state-of-the-art naive Bayes classifiers, where CAWNB+, WANBIA+, AIWNB^E+, AIWNB^L+ and RNB+ utilize the proposed SADD discretization scheme and others utilize MDLP [44].

Dataset	CAWNB	CAWNB+	WANBIA	WANBIA+	AIWNB^E	AIWNB^E+	AIWNB^L	AIWNB^L+	RNB	RNB+
Iris	93.33±5.96	96.00±3.44	93.33±5.96	96.67±3.51	92.67±6.29	96.00±4.66	92.67±6.29	96.67±3.51	93.33±5.96	**96.67±3.51**
Parkinson	84.68±5.88	89.79±6.48	85.76±6.59	90.34±6.06	79.46±5.24	84.52±6.68	81.55±6.19	86.63±5.54	85.23±6.04	**90.34±6.06**
Seeds	90.00±6.88	91.90±4.52	90.00±5.81	92.38±4.60	87.62±4.36	90.48±6.35	87.62±4.36	91.90±5.04	89.52±7.00	**92.38±4.02**
Glass	73.85±3.51	73.72±7.72	71.13±8.30	72.42±5.51	74.28±6.86	74.29±5.92	**75.28±7.75**	74.81±4.74	71.06±4.22	73.77±7.55
Heart	77.41±9.86	83.70±9.43	84.07±10.08	83.33±10.80	83.70±8.64	83.33±9.76	83.70±9.10	83.33±9.11	**84.07±9.81**	83.33±10.66
Ecoli	83.38±3.39	85.41±3.81	82.51±3.79	84.89±6.12	82.23±4.92	84.84±4.43	82.23±4.92	84.55±5.93	83.39±3.34	**85.41±3.81**
Bupa	53.27±9.52	62.51±11.86	53.27±9.52	62.51±11.86	42.02±0.84	62.24±9.24	42.02±0.84	61.67±9.71	53.27±9.52	**62.51±11.86**
Ionophere	89.23±5.10	90.63±4.56	**92.94±6.17**	91.76±5.38	89.52±4.96	90.65±5.08	90.37±4.83	92.08±5.25	91.81±6.03	90.35±5.95
Movement	68.42±5.27	74.87±4.31	67.16±3.44	74.94±8.04	64.63±4.83	75.39±7.60	67.12±4.33	75.28±6.23	65.63±6.88	**76.83±5.32**
ILPD	67.57±4.52	69.97±4.66	67.57±4.52	**69.98±4.72**	66.54±4.43	67.74±4.29	66.89±3.90	67.92±3.40	67.92±4.35	69.63±4.48
Breast	95.85±1.49	96.28±2.04	96.28±1.59	96.42±2.64	97.28±1.20	**97.57±1.66**	96.99±1.50	97.42±1.89	96.42±1.60	96.42±1.55
Pima	75.12±5.58	76.17±4.49	74.21±4.76	76.17±4.08	73.69±4.34	74.60±5.53	73.82±4.64	75.12±5.32	74.86±5.41	**76.17±4.31**
Vowel	61.11±4.99	76.26±5.54	61.62±4.78	76.77±5.59	59.70±4.18	76.97±4.06	66.97±5.11	72.63±4.80	60.91±4.47	**76.97±4.97**
Biodegradation	84.64±2.32	85.21±3.01	85.12±2.44	85.69±3.18	81.13±2.75	82.84±2.05	81.61±2.38	83.69±1.76	85.21±2.33	**85.78±3.26**
Mice Protein	98.89±1.29	99.82±0.39	99.63±0.62	99.72±0.44	97.32±1.32	99.45±0.77	98.33±1.15	99.54±0.65	99.63±0.62	**99.91±0.29**
Yeast	57.56±3.94	59.37±3.44	56.75±4.07	59.38±4.64	57.15±3.46	59.17±4.41	57.15±3.80	59.32±4.36	57.29±3.99	**59.38±3.97**
Mfeat-fac	93.85±2.14	94.80±1.90	95.55±1.40	95.90±1.07	94.20±1.77	95.10±1.54	95.15±1.49	**95.95±1.40**	95.40±1.76	95.65±0.91
Cardio	88.66±1.49	88.94±1.88	88.66±1.56	88.42±1.89	80.53±1.42	81.33±2.16	82.64±1.32	82.79±1.62	88.70±1.87	**89.13±1.50**
Madelon	63.81±3.47	64.96±3.63	63.35±3.68	64.88±3.92	61.92±3.34	64.65±3.99	61.92±3.34	64.62±3.96	62.96±3.52	**65.15±3.11**
Spambase	94.11±0.74	**94.26±1.08**	93.78±0.92	93.89±1.28	89.94±1.25	90.35±1.78	90.16±1.27	90.53±1.85	93.94±1.19	93.96±1.06
Wave	84.36±1.22	84.86±1.44	83.88±1.52	84.16±1.54	80.08±1.05	80.48±1.09	80.74±1.28	81.22±0.94	84.30±1.56	**85.08±1.19**
Wall-Following	96.52±1.57	96.56±1.75	97.42±0.57	97.49±0.63	90.95±1.31	91.90±0.83	93.49±0.84	93.53±0.59	**97.53±0.65**	97.43±0.77
Page-Block	96.38±0.65	**96.60±0.69**	96.04±0.83	96.31±0.74	93.02±1.30	93.11±1.32	93.79±1.07	94.37±1.30	96.33±0.87	96.55±0.78
Opdigit	94.73±0.94	94.95±1.18	93.45±1.04	93.83±1.03	92.30±0.43	92.60±0.64	93.11±0.37	93.31±0.70	95.23±0.81	**95.77±0.72**
Satellite	84.40±1.01	84.48±1.40	84.52±0.76	85.00±0.93	81.69±1.26	81.88±1.62	85.33±1.00	85.72±1.33	85.81±0.92	**86.37±0.91**
Wine	51.70±1.21	52.20±1.36	53.19±1.37	53.63±1.94	48.99±1.38	50.33±1.03	50.64±1.35	51.50±1.26	53.36±1.66	**53.70±1.10**
Musk	97.24±0.73	**97.33±0.44**	96.23±0.87	96.15±0.40	92.91±0.86	92.89±0.81	93.79±0.93	93.66±0.64	95.89±0.68	97.04±0.71
Anuran	95.41±0.68	95.55±0.84	94.66±0.57	95.11±0.60	88.87±1.07	89.46±1.18	92.93±1.08	93.58±1.08	95.44±0.68	**95.97±0.75**
Pendigit	93.06±0.42	93.76±0.62	89.71±0.76	90.04±0.76	88.72±1.11	89.24±1.22	93.41±0.68	93.72±0.71	93.15±0.35	**93.81±0.71**
Magic	83.43±0.75	**83.61±0.65**	82.40±0.63	82.47±0.77	79.32±0.49	79.81±0.54	80.18±0.62	80.71±0.38	83.40±0.72	83.41±0.83
IndoorLoc	**87.09±1.53**	86.30±3.92	86.30±0.66	86.52±0.76	65.27±1.26	68.56±1.13	68.08±0.96	68.66±1.23	83.59±2.99	86.64±0.96
AVG	82.55	84.54	82.60	84.42	79.28	81.99	80.63	82.79	82.73	84.89

AIWNB^E, AIWNB^L and RNB, respectively, among which the results on 9, 6, 14, 9, and 14 datasets are statistically significant.

TABLE 4.10: Summary of statistical significance tests of the proposed SADD over MDLP [44] on various NB classifiers. For each entry $u(v)$, u is the number of datasets on which CAWNB+, WANBIA+, AIWNB^E+, AIWNB^L+ and RNB+ outperform their counterparts, and v is the number of datasets on which the performance gain is statistically significant with the significance level of $p = 0.05$.

CAWNB+ vs. CAWNB	WANBIA+ vs. WANBIA	AIWNB^E+ vs. AIWNB^E	AIWNB^L+ vs. AIWNB^L	RNB+ vs. RNB
28(9)	27(6)	29(14)	19(9)	28(14)

4.5 Summary

In this paper, we aim to design a discretization and classification framework to balance the generalization capability and discrimination power, during both data discretization and classification. We find that a popular discretization scheme, MDLP, often results in an early stop during the top-down discretization, which leads to a huge information loss. To address this problem, we propose a semi-supervised adaptive discriminative discretization (SADD) method, which utilizes the pseudo-labeling technique to make full use of the discriminant information residing in both labeled and unlabeled data. Furthermore, an adaptive discriminative discretization scheme is designed to resolve the problem of huge information loss in MDLP. In such a way, the proposed SADD retains the discriminant information for the classifier while preserving its generalization ability. Besides, the proposed RNB+ well balances the generalization ability and discrimination power during both data discretization and feature weighting. Experimental results on 31 machine-learning datasets demonstrate that the proposed SADD significantly outperforms all compared discretization methods and the proposed RNB+ significantly outperforms other state-of-the-art NB classifiers.

Chapter 5

A Max-relevance-min-divergence Criterion for Data Discretization with Applications on Naive Bayes

In many classification models, data is discretized to better estimate its distribution. Existing discretization methods often target at maximizing the discriminant power of discretized data, while overlooking the fact that the primary target of data discretization in classification is to improve the generalization performance[1]. As a result, the data tend to be over-split into many small bins since the data without discretization retain the maximal discriminant information. In this Chapter, we propose a Max-Dependency-Min-Divergence (MDmD) criterion that maximizes both the discriminant information and generalization ability of the discretized data. More specifically, the Max-Dependency criterion maximizes the statistical dependency between the discretized data and the classification variable while the Min-Divergence criterion explicitly minimizes the JS-divergence between the training data and the validation data for a given discretization scheme. The proposed MDmD criterion is technically appealing, but it is difficult to reliably estimate the high-order joint distributions of attributes and

[1] The work has been published in Pattern Recognition[196].

the classification variable. We hence further propose a more practical solution, Max-Relevance-Min-Divergence (MRmD) discretization scheme, where each attribute is discretized separately, by simultaneously maximizing the discriminant information and the generalization ability of the discretized data. The proposed MRmD is compared with the state-of-the-art discretization algorithms under the naive Bayes classification framework on 45 machine-learning benchmark datasets. It significantly outperforms all the compared methods on most of the datasets.

5.1 Introduction

Deep-learning models have been successful in many applications [8], but they require a large amount of training samples. For applications such as drug discovery [197] and medical diagnosis [181], it is labor-expensive to collect many samples, where traditional machine-learning methods with much fewer model parameters may generalize better, *e.g.*, decision tree [198], fuzzy rule-based classifiers [79], naive Bayes [9, 20, 22], k-nearest-neighbor classifier [129], and support vector machine [45, 132]. To improve the generalization capability, and to handle mixed-type data, continuous attributes are often discretized to facilitate a better estimation of the data distribution for subsequent classifiers [78, 81, 83, 132]. The discrete features are easier to understand than continuous ones because they are closer to knowledge-level representation [45]. Most importantly, by discretizing similar values into one bin, the distribution discrepancy between training data and test data could be reduced, and hence the generalization capability of a classifier could be enhanced [9, 20, 22].

Data discretization aims to find a minimal set of cut points that optimally discretize continuous attributes to maximize the classification accuracy [45, 83]. Existing methods often over-emphasize maximizing the discriminant information, while neglecting the primary target of data discretization, *i.e.*, to improve the generalization performance by reducing the noisy information that is harmful to reliable classification. In data discretization, two opposing goals often compete with each other, *i.e.*, the generalization performance is maximized when all samples are discretized into one bin so that there is no distribution discrepancy between training data and test data, but the discriminant information is totally lost in this case. On the other hand, the discriminant information is maximized when no discretization is performed on the data, but the generalization performance would not be improved.

In literature, many discretization algorithms have been developed to maximize the dependence between discrete attributes and classification variables in terms of mutual information [81], information entropy [44, 80], contingency coefficient [25, 113], statistical interdependency [23, 24], and many others [45, 83, 102, 128]. However, none of them explicitly maximizes the generalization capability. Instead, they often restrict the number of intervals after discretization to be small, in the hope of retaining the generalization ability, *e.g.*, Class-Attribute Interdependence Maximization (CAIM) [23] and Class-Attribute Contingency Coefficient (CACC) [25] both restrict the number of intervals to the number of classes. Such a design does not optimize the discretization scheme in terms of the generalization.

To tackle this problem, a Max-Dependency-Min-Divergence (MDmD) criterion is proposed to simultaneously maximize the discriminant power and the generalization ability. The Max-Dependency criterion maximizes the mutual information between the discrete data and the classification variable [67, 81]. It has been widely used in feature selection [67, 134, 199], feature weighting [20] and feature extraction [172]. Regarding the generalization ability, existing discretizers often choose to maintain a small number of intervals [24, 45, 83]. However, if the number of intervals is too small, a significant amount of discriminant information will be lost. On the other hand, if it is too large, the resulting discretization scheme may not generalize well to the test data. It is hence difficult to decide the optimal number of intervals. In this paper, a Min-Divergence criterion is proposed to explicitly maximize the generalization ability by minimizing the divergence between the distribution of training data and that of validation data. This criterion is integrated with the Max-Dependency criterion to form the proposed MDmD criterion, which could achieve a better trade-off between the discriminant power and generalization ability so that the subsequent classifier could work well.

The proposed MDmD criterion is technically appealing but difficult to be applied in practice, as it is difficult to reliably estimate the high-order joint distributions between attributes and classification variable. To tackle this problem, instead of maximizing the mutual information between all attributes and the classification variable, we propose to maximize the summation of the mutual information between each attribute and the classification variable, also known as Max-Relevance [67, 81]. At the same time, we propose to minimize the summation of divergences between distributions when one attribute is evaluated at a time. These two criteria are combined to form the proposed Max-Relevance-Min-Divergence (MRmD) criterion, which maximizes the discriminant power by maximizing the mutual information between discrete attributes and the classification variable, and simultaneously maximizes the generalization ability by minimizing the divergence between the distributions of training data and validation data. It is time-consuming to exhaustively search for the global optimal solution. Following the design of many discretization methods, *e.g.*, MDLP [44], CAIM [23] and Ameva [113], a greedy top-down hierarchical splitting algorithm is used together with the proposed MRmD criterion to derive a near-optimal discretization scheme.

The proposed MRmD criterion is integrated with one of the most recent developments of naive Bayes classifier, Regularized Naive Bayes (RNB) [9], and compared with the state-of-the-art discretization methods and classifiers on 45 benchmark datasets. The proposed method significantly outperforms the compared methods on most of the datasets.

Our contributions can be summarized as follows. 1) We identify the key limitations of existing discretization methods that they often overemphasize maximizing the discriminant power, which limits the improvement of the generalization ability. 2) To tackle this problem, a Max-Dependency-Min-Divergence criterion is proposed to simultaneously maximize the discriminant power and minimize the distribution discrepancy so that the derived discretization scheme could generalize well to the data population. 3) To tackle the challenges of reliable estimation of the joint probabilities in MDmD, a more practical solution, Max-Relevance-Min-Divergence discretization scheme, is proposed to derive the optimal discretization scheme for one attribute at a time. 4) The proposed method is systematically evaluated on 45 benchmark datasets and demonstrates superior performance compared with the state-of-the-art discretization methods and classifiers.

5.2 Related work

Discretization methods have been deployed to extract knowledge from data in many machine learning algorithms such as decision tree [198], rule-based learning [79] and naive Bayes [9, 20, 22]. Discretization methods can be categorized according to many properties [45]:

Local vs. Global: Local methods [44, 80] generate intervals based on partial data, whereas global ones [23–25, 45] consider all available data.

Dynamic vs. Static: Dynamic discretizers [81] interact with learning models whereas static ones [24, 82] execute before the learning stage.

Splitting vs. Merging: This relates to the top-down split [23, 25, 80] or bottom-up merge [82] strategy in producing new intervals.

Univariate vs. Multivariate: Univariate algorithms [24, 78, 80] discretize each attribute separately whereas multivariate discretizers [45, 83] consider a combination of attributes when discretizing data.

Direct vs. Incremental: Direct methods [45, 84] divide the range into several intervals simultaneously, while incremental ones [23–25, 44, 80] begin with a simple discretization and improve it gradually using more criteria.

Depending on whether the class label is used, discretization methods can be divided into supervised, semi-supervised and unsupervised methods [45]. Equal-width and equal-frequency discretization are representative unsupervised methods [78]. Minimal Optimized Description Length is a representative semi-supervised method, which applies the Bayesian rule on both labeled and unlabeled data for discretization [127]. Supervised methods can be further divided into wrapper-based methods [45, 83, 128–130] and filter-based methods [23–25, 80, 82]. The former optimizes the discretization scheme by utilizing the classification feedback [45, 83, 128–130], while the latter optimizes some indirect target for data discretization, *e.g.*, information entropy [44, 80], mutual information [81] and interdependency [23–25].

Wrapper-based methods [45, 83, 128–130] iteratively refine the discretization scheme by using the classification feedback. Evolutionary algorithms are often utilized to discretize data by maximizing the classification accuracy and minimizing the number of intervals [45]. Tahan and Asadi developed an evolutionary multi-objective discretization to handle the imbalanced datasets [83]. Tran *et al.* initialized the discretization scheme by using the MDLP criterion [44] and utilized barebones particle swarm optimization to fine-tune the derived scheme [129]. In [128], the particle swarm optimization strategy is used to explore the interaction between features for better discretization. Chen *et al.* developed a genetic algorithm based on the fuzzy rough set to effectively explore the data association [130].

Filter-based methods [23, 25, 80, 82] have been popular in recent years for their strong theoretical background. MDLP is one of the most popular discretization methods in many classifiers [9, 20, 22], which hierarchically partitions data by maximizing the information entropy [44]. To avoid excessive splitting, it defines a stop criterion derived from channel coding theory. Xun *et al.* developed a multi-scale discretization method to obtain the set of cut points with different granularity and utilized the MDLP criterion to determine the best cut point [80]. Other statistical measures have also been widely used in data discretization [24, 78, 82]. Kurgan and Cios developed a CAIM criterion based on a quanta matrix to select boundary points iteratively within a pre-defined number of intervals [23]. Cano *et al.* extended it for multi-label data [24]. Tsai *et al.* introduced a discretization method based on CACC by taking the overall data distribution into account [25]. In [132], low-frequency values are discretized and the correlation between discrete attribute and continuous attribute is utilized for discretization. Chi-square statistics between the discrete data and the classification variable, *e.g.*, modified Chi2 and extended Chi2, have been recently developed for data discretization [82].

Most discretization methods [23–25, 44, 80] emphasize maximizing the discriminant power, but they pay little attention to the generalization capability, *e.g.*, they often restrict the number of discrete intervals to be small, in the hope of achieving a satisfactory generalization ability. If a discretization method considers maximizing these two simultaneously, the subsequent classifier will achieve a better classification performance on novel test data.

5.3 Proposed discretization method

5.3.1 Analysis of existing discretization methods

Data discretization is crucial to improve the learning efficiency and generalization of classifiers [45]. Many methods have been designed to maximize the statistical dependency between discretized features and classification variables in many different forms, *e.g.*, information gain in MDLP [44, 80], class-attribute dependency in CADD [91], and class-attribute interdependency in CAIM [23, 24]. But they often neglect the fact that the primary target of data discretization in classification is to improve generalization ability, *i.e.*, by discretizing similar values into one interval, the data distribution can be better estimated so that it fits well to novel test samples.

MDLP is one of the most widely used discretization methods [9, 20, 22], *e.g.*, it is the default discretization method in Weka toolbox [200]. It hierarchically splits the dynamic range into smaller ones. For each attribute, a cut point d is selected to divide the current set $\mathcal{S}$ into two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$, which maximizes the information gain $G(\mathcal{S}, d) = E(\mathcal{S}) - \frac{|\mathcal{S}_1|}{|\mathcal{S}|}E(\mathcal{S}_1) - \frac{|\mathcal{S}_2|}{|\mathcal{S}|}E(\mathcal{S}_2)$, where $E(\mathcal{S}) = -\sum\limits_{c\in\mathcal{C}} P(c, \mathcal{S}) \log P(c, \mathcal{S})$ is the entropy, $P(c, \mathcal{S})$ is the probability of class c in $\mathcal{S}$ and $\mathcal{C}$ is the set of classes. It can be shown that the accumulative information gain is equivalent to the mutual information between the discrete attribute and the classification variable. Greedily maximizing the information gain may split the attribute into too many small intervals with too few samples so that the likelihood probabilities can not be reliably estimated. To prevent this, MDLP requires $G(\mathcal{S}, d)$ to be greater than a threshold that is derived from the overhead of information encoding, which may not be in line with the classification point of view. It often leads to an early stop during splitting, and hence a significant discriminant information loss.

Class-Attribute Dependent Discretizer (CADD) [91] maximizes the discriminant information via maximizing $CADD(X, C) = \frac{I(X;C)}{E(X,C)}$, where $I(X; C)$ is the mutual information and $E(X, C)$ is the joint entropy. Maximizing CADD tends to produce too many small discretization intervals. To prevent this, a user-specified threshold is utilized to constrain the number of intervals, but with no guarantee of optimality.

The CAIM discretization utilizes a heuristic measure $CAIM(X,C) = \frac{1}{n}\sum_{i=1}^{n} \frac{\max_{c \in \mathcal{C}} q_{i,c}^2}{M_i}$ to model the interdependence between classes and attributes [23, 24], where $q_{i,c}$ is the number of samples in class c in the i-th interval, and M_i is the number of samples in the i-th interval. The number of intervals generated by CAIM is often close to the number of classes, which may limit the performance of CAIM, especially when the number of classes is small.

Existing approaches mainly focus on maximizing the discriminant power, which often split attributes into too many small intervals. This defeats the purpose of data discretization in classification, *i.e.*, to improve the generalization. To retain the generalization ability, they often restrict the number of intervals to a predefined number or the number of classes [23–25], or require the information gain to be larger than a threshold [44, 80]. These methods lack a measure to explicitly maximize the generalization ability.

5.3.2 Maximal-dependency-minimal-divergence for data discretization

In this paper, the target is to derive an optimal discretization scheme $\boldsymbol{\mathcal{D}}$ to transform continuous attributes $\boldsymbol{X}$ into discrete ones $\boldsymbol{A}$, which simultaneously maximizes the discriminant power of the discretized data and maximizes the generalization ability to the data not used in training,

$$\boldsymbol{A} = f_D(\boldsymbol{X}, \boldsymbol{\mathcal{D}}), \tag{5.1}$$

where f_D denotes the discretization function, and $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m\}$ contains the discretization schemes for m features.

5.3.2.1 Maximal-dependency criterion

To maximize the discriminant information, we propose to maximize the mutual information $I(\boldsymbol{A}; C)$ between the discretized attributes $\boldsymbol{A}$ and the classification variable C given the discretization scheme $\boldsymbol{\mathcal{D}}$,

$$\boldsymbol{\mathcal{D}}^* = \text{argmax}_{\boldsymbol{\mathcal{D}}}\, I(\boldsymbol{A}; C), \tag{5.2}$$

$$I(\boldsymbol{A};C) = \sum_{c \in \mathcal{C}} \sum_{\boldsymbol{a} \in \boldsymbol{A}} P(\boldsymbol{a}, c) \log \frac{P(\boldsymbol{a}, c)}{P(\boldsymbol{a})P(c)}, \tag{5.3}$$

where $P(\boldsymbol{a}, c)$ is the joint probability distribution and $P(\boldsymbol{a})$ and $P(c)$ are the respective marginal probabilities. This criterion is often known as Max-Dependency [67, 81]. Apparently, it is difficult to reliably estimate both the joint probability distribution $P(\boldsymbol{a}, c)$ and the marginal probability distribution $P(\boldsymbol{a})$ due to the high dimensionality. Furthermore, the Max-Dependency criterion tends to greedily maximize the discriminant power and hence over-discretize the continuous data into too many small intervals, *i.e.*, each unique value in the numerical attribute may be treated as a separate interval, in which the generalization capability would not be improved.

5.3.2.2 Minimal-divergence criterion

To maximize the generalization ability, we propose to minimize the Jensen-Shannon (JS) divergence [201] between the training data distribution and the test data distribution. As the latter is in general unknown, we hence aim to minimize the distribution discrepancy between training data and validation data instead. The intuition behind is that by minimizing the JS divergence $D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a}))$ describing the similarity between the distribution $P^t(\boldsymbol{a})$ of the training data $\boldsymbol{A}^t$ and the distribution $P^v(\boldsymbol{a})$ of the validation data $\boldsymbol{A}^v$, the derived discretization scheme $\boldsymbol{\mathcal{D}}$ could generalize well from the training data to the novel test data. Formally, $D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a}))$ is defined as:

$$D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a})) = \frac{1}{2}(D_{KL}(P^t(\boldsymbol{a})||P^*(\boldsymbol{a})) + D_{KL}(P^v(\boldsymbol{a})||P^*(\boldsymbol{a}))), \tag{5.4}$$

where $P^*(\boldsymbol{a}) = \frac{1}{2}(P^t(\boldsymbol{a}) + P^v(\boldsymbol{a}))$ and $D_{KL}(P^t(\boldsymbol{a})||P^*(\boldsymbol{a}))$ is the Kullback-Leibler divergence between $P^t(\boldsymbol{a})$ and $P^*(\boldsymbol{a})$,

$$D_{KL}(P^t(\boldsymbol{a})||P^*(\boldsymbol{a})) = \sum_{\boldsymbol{a} \in \mathcal{A}} P^t(\boldsymbol{a}) \log \frac{P^t(\boldsymbol{a})}{P^*(\boldsymbol{a})}. \tag{5.5}$$

Similarly, $D_{KL}(P^v(\boldsymbol{a})||P^*(\boldsymbol{a}))$ is defined as:

$$D_{KL}(P^v(\boldsymbol{a})||P^*(\boldsymbol{a})) = \sum_{\boldsymbol{a} \in \mathcal{A}} P^v(\boldsymbol{a}) \log \frac{P^v(\boldsymbol{a})}{P^*(\boldsymbol{a})}. \tag{5.6}$$

$P^t(\boldsymbol{a})$, $P^v(\boldsymbol{a})$ and $P^*(\boldsymbol{a})$ are the probability distributions of $\boldsymbol{a}$ given the attribute set $\boldsymbol{A}^t$, $\boldsymbol{A}^v$ and $\boldsymbol{A}^*$ respectively. The JS divergence has been utilized as a distance metric between two distributions. It is symmetric, *i.e.*, $D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a})) = D_{JS}(P^v(\boldsymbol{a})||P^t(\boldsymbol{a}))$. $D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a})) \in [0, 1]$. The smaller JS divergence represents the higher similarity between these two distributions, and hence the derived discretization scheme could generalize well to the novel test data. The Minimal-Divergence criterion is hence defined as:

$$\boldsymbol{\mathcal{D}}^* = \text{argmin}_{\boldsymbol{\mathcal{D}}}\, D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a})). \tag{5.7}$$

5.3.2.3 Maximal-dependency-minimal-divergence criterion

To simultaneously maximize the discriminant power and the generalization ability of the discretized attributes, we propose to maximize the dependency $I(\boldsymbol{A}; C)$ between discrete attributes $\boldsymbol{A}$ and classification variable C, and minimize the divergence $D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a}))$ between the distribution of training data and that of validation data given the discretization scheme $\boldsymbol{\mathcal{D}}$,

$$\boldsymbol{\mathcal{D}}^* = \text{argmax}_{\boldsymbol{\mathcal{D}}}\, \lambda I(\boldsymbol{A}; C) - D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a})), \tag{5.8}$$

where λ is the hyper-parameter to balance the two terms. Note that the two terms compete with each other. On the one hand, when all the data are discretized into one bin, $I(\boldsymbol{A}; C) = 0$, indicating that the discriminant information is totally lost, but $D_{JS}(P^t(\boldsymbol{a})||P^v(\boldsymbol{a})) = 0$, *i.e.*, the two distributions are identical, and hence the generalization is maximized. On the other hand, when each unique sample is discretized into a separate bin, the discriminant information is maximized, while it does not improve the generalization ability. The proposed MDmD criterion provides a solution to find the optimal trade-off between the discriminant power and the generalization ability.

5.3.3 Maximal-relevance-minimal-divergence criterion for data discretization

The proposed MDmD criterion is technically appealing but difficult to implement in practice, as it is hard to reliably estimate the high-order joint distribution $P(\boldsymbol{a}, c)$,

$P^t(\boldsymbol{a})$ and $P^v(\boldsymbol{a})$. Inspired by [67], we propose the Max-Relevance criterion for data discretization. More specifically, following the chain rule of mutual information [201], $I(\boldsymbol{A};C) = \sum_{j=1}^{m} I(A_j;C|A_{j-1},\cdots,A_1)$, where $I(A_j;C|A_{j-1},\cdots,A_1)$ is the conditional mutual information between A_j and C conditioned on $A_{j-1},\cdots,A_1$. If we ignore the high-order interaction between features, *i.e.*, $I(A_j;C|A_{j-1},\cdots,A_1) \approx I(A_j;C)$, we have $I(\boldsymbol{A};C) \approx \sum_{j=1}^{m} I(A_j;C)$, where $I(A_j;C)$ is the mutual information between A_j and C. The detailed derivation of this approximation can be found in [201]. The Max-Relevance criterion is then given as follows,

$$\boldsymbol{\mathcal{D}}^* = \mathrm{argmax}_{\boldsymbol{\mathcal{D}}} \sum_{j=1}^{m} I(A_j;C), \tag{5.9}$$

which has been widely used to approximate the Max-Dependency criterion [67, 81, 199].

For the second term in Eqn. (5.8), instead of estimating the divergence between $P^t(\boldsymbol{a})$ and $P^v(\boldsymbol{a})$ jointly considering all the attributes, $D_{JS}(P^t(\mathbf{a})||P^v(\mathbf{a}))$ can be estimated by considering them one by one. Following the chain rule of divergence [201],

$$D_{KL}(P^t(\mathbf{a})||P^*(\mathbf{a})) = \sum_{j=1}^{m} D_{KL}(P^t(a_j|a_{j-1},\cdots,a_1)||P^*(a_j|a_{j-1},\cdots,a_1)), \tag{5.10}$$

where $D_{KL}(P^t(a_j|a_{j-1},\cdots,a_1)||P^*(a_j|a_{j-1},\cdots,a_1))$ is the conditional divergence. If we ignore the high-order interaction between features, *i.e.*, $P^t(a_j|a_{j-1},\cdots,a_1) \approx P^t(a_j)$ and $P^*(a_j|a_{j-1},\cdots,a_1) \approx P^*(a_j)$, we have $D_{KL}(P^t(\mathbf{a})||P^*(\mathbf{a})) \approx \sum_{j=1}^{m} D_{KL}(P^t(a_j)||P^*(a_j))$. It is then easy to show that $D_{JS}(P^t(\mathbf{a})||P^v(\mathbf{a})) \approx \sum_{j=1}^{m} D_{JS}(P^t(a_j)||P^v(a_j))$, where $D_{JS}(P^t(a_j)||P^v(a_j))$ is the JS divergence between training data distribution and validation data distribution for the j-th attribute given the discretization scheme $\mathcal{D}_j$. The Min-Divergence criterion can hence be simplified as,

$$\boldsymbol{\mathcal{D}}^* = \mathrm{argmin}_{\boldsymbol{\mathcal{D}}} \sum_{j=1}^{m} D_{JS}(P^t(a_j)||P^v(a_j)), \tag{5.11}$$

We combine Eqn. (5.9) and Eqn. (5.11) to form the proposed Maximal-Relevance-Minimal-Divergence (MRmD) criterion,

$$\boldsymbol{\mathcal{D}}^{*} = \mathrm{argmax}_{\boldsymbol{\mathcal{D}}} \sum_{j=1}^{m} \left[\lambda I(A_j; C) - D_{JS}(P^{t}(a_j)||P^{v}(a_j))\right]. \tag{5.12}$$

Given the discretization scheme $\boldsymbol{\mathcal{D}}$, each original feature $\boldsymbol{x}_j$ can be discretized into A_j as in Eqn. (5.1), and then the mutual information $I(A_j; C)$ and JS divergence $D_{JS}(P^{t}(a_j)||P^{v}(a_j))$ can be estimated. The proposed MRmD aims to find the optimal scheme $\boldsymbol{\mathcal{D}}^{*}$ that maximizes the relevance of discretized attributes with respect to the classification variable via the first term, and maximizes the generalization ability by minimizing the distribution discrepancy between training data and validation data via the second term.

To derive $\boldsymbol{\mathcal{D}}^{*}$, it is not difficult to show that each attribute A_j can be processed separately to derive its optimal discretization scheme $\mathcal{D}_j^{*}$,

$$\mathcal{D}_j^{*} = \mathrm{argmax}_{\mathcal{D}_j}\, \Psi(A_j; C), \tag{5.13}$$

$$\Psi(A_j; C) = \lambda I(A_j; C) - D_{JS}(P^{t}(a_j)||P^{v}(a_j)). \tag{5.14}$$

After deriving the optimal solution for each attribute, the optimal discretization scheme is obtained as $\boldsymbol{\mathcal{D}}^{*} = \{\mathcal{D}_1^{*}, \mathcal{D}_2^{*}, ..., \mathcal{D}_m^{*}\}$.

5.3.4 Proposed MRmD discretization

An MRmD discretization method is proposed to discretize the attributes one at a time, with the block diagram shown in Fig. 6.1. For each continuous attribute, the proposed method iteratively determines the cut points for discretization by maximizing the MRmD criterion, where the Max-Relevance criterion is achieved by maximizing the mutual information between the discretized attribute and the classification variable, and the Min-Divergence criterion is achieved by minimizing the distribution discrepancy between training data and validation data. The two criteria are combined as the MRmD criterion, to derive an optimal set of cut points to discretize the attributes.

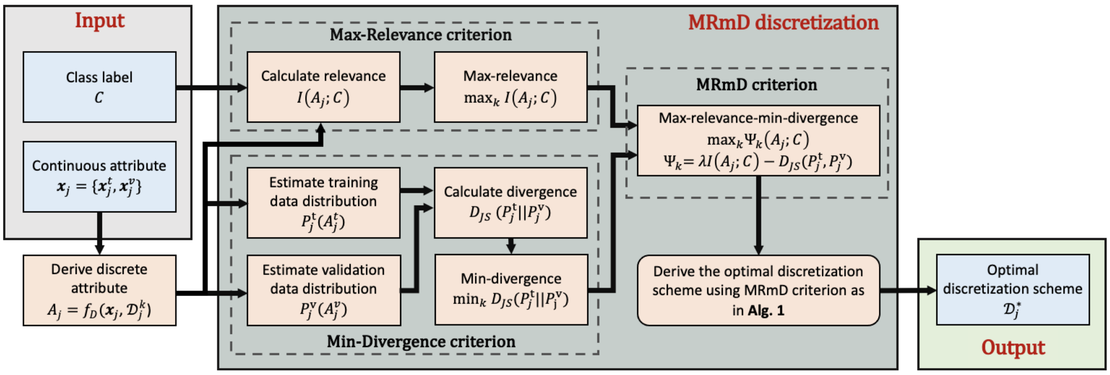

Input
Class label
C
Continuous attribute
$\boldsymbol{x}_j = \{\boldsymbol{x}_j^t, \boldsymbol{x}_j^v\}$
Derive discrete attribute
$A_j = f_D(\boldsymbol{x}_j, \mathcal{D}_j^k)$
MRmD discretization
Max-Relevance criterion
Calculate relevance
$I(A_j; C)$
Max-relevance
$\max_k I(A_j; C)$
Estimate training data distribution
$P_j^t(A_j^t)$
Estimate validation data distribution
$P_j^v(A_j^v)$
Calculate divergence
$D_{JS}(P_j^t||P_j^v)$
Min-divergence
$\min_k D_{JS}(P_j^t||P_j^v)$
Min-Divergence criterion
MRmD criterion
Max-relevance-min-divergence
$\max_k \Psi_k(A_j; C)$
$\Psi_k = \lambda I(A_j; C) - D_{JS}(P_j^t, P_j^v)$
Derive the optimal discretization scheme using MRmD criterion as in Alg. 1
Optimal discretization scheme
$\mathcal{D}_j^*$
Output

FIGURE 5.1: The proposed MRmD discretization framework. The optimal set of cut points for each attribute is selected by maximizing the proposed MRmD criterion. More details are given in Algo. 4.

Following the design in MDLP [44, 80] and many others [23–25], a greedy top-down hierarchical splitting paradigm is designed to derive the optimal solution. More specifically, for each attribute $\boldsymbol{x}_j$, we initialize its discretization scheme $\mathcal{D}_j^*$ as an empty set, and treat the whole dynamic range initially as one interval. Hence $I(A_j; C) = 0$ and $D_{JS}(P^t(a_j)||P^v(a_j)) = 0$. The optimal cut points $\mathcal{D}_j^*$ are selected from a candidate set $\mathcal{S}_j$, which is initialized as $\mathcal{U}(\boldsymbol{x}_j)$, the unique values of $\boldsymbol{x}_j$. It can be shown that the number of possible discretization schemes for $\boldsymbol{x}_j$ is $2^{|\mathcal{S}_j|}$. It is expensive to exhaustively evaluate every feasible discretization scheme. $\forall d_k \in \mathcal{S}_j$, $\mathcal{D}_j^k = \mathcal{D}_j^* \cup d_k$, and we use $\mathcal{D}_j^k$ to discretize $\boldsymbol{x}_j$, and evaluate the MRmD criterion Ψ_k defined in Eqn. (5.14) for every d_k. Then, we select the cut point $d_{k_{max}}$ that maximizes Ψ_k, and update the optimal discretization scheme as $\mathcal{D}_j^* = \mathcal{D}_j^* \cup d_{k_{max}}$. The candidate set is updated as $\mathcal{S}_j = \mathcal{S}_j - d_{k_{max}}$. The proposed method incrementally selects the cut point to divide the dynamic range into intervals until the criterion defined in Eqn. (5.14) does not increase anymore. The proposed MRmD discretization is summarized in Algo. 4. the proposed MRmD discretizes the continuous attribute $\boldsymbol{x}_j \in \boldsymbol{X}$ one by one. These discretization schemes $\mathcal{D}_j^*$ for all attributes form the complete discretization scheme $\boldsymbol{\mathcal{D}}^*$. The proposed MRmD generates a discretization scheme that simultaneously maximizes the discriminant information and the generalization ability, and hence improves the classification performance.

Algorithm 4 The proposed MRmD discretization scheme.

```
Input: Input data $\boldsymbol{X} = \{\boldsymbol{X}^t, \boldsymbol{X}^v\}$ with class label $\boldsymbol{c}$ for $\boldsymbol{X}$
Output: Discretization scheme $\boldsymbol{\mathcal{D}}^* = \{\mathcal{D}_1^*, \mathcal{D}_2^*, ..., \mathcal{D}_m^*\}$
$\boldsymbol{\mathcal{D}}^* \leftarrow \emptyset$    ▷ Initialize $\boldsymbol{\mathcal{D}}^*$ as an empty set
for $\boldsymbol{x}_j \in \boldsymbol{X}$ do    ▷ Loop through all attributes
    $\mathcal{D}_j^* \leftarrow \emptyset$    ▷ Initialize $\mathcal{D}_j^*$ as an empty set
    $\mathcal{S}_j \leftarrow \mathcal{U}(\boldsymbol{x}_j)$    ▷ Initialize $\mathcal{S}_j$ as the set of unique values
    $\Psi_{max} \leftarrow -\infty$    ▷ Initialize optimal MRmD value
    while $(\mathcal{S}_j \neq \emptyset)$ do    ▷ Incrementally select the cut point
        for $d_k \in \mathcal{S}_j$ do    ▷ For each possible cut point
            $\mathcal{D}_j^k = \mathcal{D}_j^* \cup d_k$    ▷ Include $d_k$ as the cut point
            $A_j = f_D(\boldsymbol{x}_j, D_j^k)$    ▷ Discretize $\boldsymbol{x}_j$ using $\mathcal{D}_j^k$
            $\Psi_k = \lambda I(A_j; C) - D_{JS}(P^t(a_j)||P^v(a_j))$    ▷ Calculate $\Psi_k$
        end for
        $k_{max} = \text{argmax}_k \Psi_k$    ▷ Derive $d_{k_{max}}$ with maximal $\Psi_{k_{max}}$
        if $\Psi_{k_{max}} \leq \Psi_{max}$ then
            break;
        end if
        $\Psi_{max} \leftarrow \Psi_{k_{max}}$    ▷ Update $\Psi_{max}$
        $\mathcal{D}_j^* \leftarrow \mathcal{D}_j^* \cup d_{k_{max}}$    ▷ Update $\mathcal{D}_j^*$
        $\mathcal{S}_j \leftarrow \mathcal{S}_j - d_{k_{max}}$    ▷ Update $\mathcal{S}_j$
    end while
    $\boldsymbol{\mathcal{D}}^* \leftarrow \boldsymbol{\mathcal{D}}^* \cup \mathcal{D}_j^*$    ▷ Add $\mathcal{D}_j^*$ into $\boldsymbol{\mathcal{D}}^*$
end for
return $\boldsymbol{\mathcal{D}}^*$
```

5.3.5 Analysis of hyper-parameter λ

The hyper-parameter λ plays an important role in the proposed MRmD. As discussed early, the MRmD value $\Psi(A_j; C)$ is initialized as zero at the beginning of the top-down discretization. Both terms in Eqn. (5.14) increases with the number of discretization intervals. It is hence difficult to derive the optimal MRmD value. To tackle this problem, λ is defined as,

$$\lambda = e^{-\frac{|\mathcal{D}_j^*|}{N_D}}, \tag{5.15}$$

where $|\mathcal{D}_j^*|$ is the number of cut points in the current discretization scheme, and N_D is empirically set to 50. The designed weighting function satisfies the following properties: 1). The value of λ is between 0 and 1. 2) λ monotonically decreases with the number of cut points. In the earlier stage, when there are only a small number of cut points in the discretization scheme, λ is large and hence more emphasis is put on the first term in Eqn. (5.14), to highlight the importance of maximizing the discriminant information.

As more cut points are added into the discretization scheme, λ becomes smaller and then more emphasis is put on the second term of Eqn. (5.14) so that more emphasis is put on improving the generalization performance. Such a design could help the proposed MRmD discretization generate a discretization scheme that achieves an optimal trade-off between the generalization capability and the discrimination information for the discretized data.

5.4 Experimental results

5.4.1 Experimental settings

The proposed MRmD discretization is compared with five popular filter-based discretization methods, Ameva [113], CAIM [23], MDLP [44], Modified Chi2 [102] and PKID [84], and a recent wrapper-based approach, EMD [45]. Ameva [113] and CAIM [23] are two popular methods for discretization in credit scoring models for operational research [202]. MDLP [44] and PKID [84] are widely used in NB classifiers [9, 20, 22] and feature selection [129]. Modified Chi2 [102] has been recently used to improve the ensemble classification methods [82]. EMD [45] has been used in many applications recently, *e.g.*, high-resolution remote sensing [130] and feature selection [128]. Two classifiers are used for evaluation, naive Bayes classifier and decision tree (C4.5) [90].

The proposed MRmD is then integrated with one of the recent naive Bayes classifiers, RNB (regularized naive Bayes) [9], denoted as MRmD-RNB. It is compared with state-of-the-art NB classifiers including RNB [9], WANBIA [21], CAWNB [22] and AIWNB [20]. It is also compared with three deep-learning models, ResNet [203], FTT [8] and PWedRVFL [204]. ResNet and WPedRVFL are implemented using the codes provided by the authors of [204], and FTT is implemented following [8].

The experiments are conducted on 45 benchmark datasets in various fields including healthcare, biology, disease diagnosis and business. The datasets are extracted from the UCI machine learning repository[2], which have been widely used to evaluate discretization algorithms [45, 78, 81] and naive Bayes classifiers [9, 20, 22]. Most datasets are collected

[2] https://archive.ics.uci.edu/ml/index.php

Table 5.1: Description of compared methods: six discretization methods, four state-of-the-art naive Bayes classifiers and three deep-learning models.

Discretization methods	
Ameva [113, 202]	Filter-based statistical top-down discretization, maximizing the contingency coefficient based on Chi-square statistics.
CAIM [23, 24]	Filter-based statistical top-down discretization, heuristically maximizing the class-attribute interdependency by using quanta matrix.
MDLP [9, 44, 80]	Filter-based entropy-based top-down discretization, maximizing the information gain using the minimum description length principle.
Modified Chi2 [82, 102]	Filter-based statistical bottom-up discretization, merging intervals dynamically by using the rough set theory.
PKID [84]	Filter-based unsupervised discretization, adjusting the number and size of intervals proportional to the number of training instances.
EMD [45, 83]	Wrapper-based multivariate discretization, minimizing the classification error and the number of intervals using genetic algorithm.
Naive Bayes classifiers	
RNB [9]	Wrapper-based attribute-weighting naive Bayes, simultaneously optimizing class-dependent and class-independent weights by using the L-BFGS algorithm.
WANBIA [21]	Wrapper-based attribute-weighting naive Bayes, optimizing class-independent attribute weights by using the L-BFGS algorithm.
CAWNB [22]	Wrapper-based attribute-weighting naive Bayes, optimizing class-specific attribute weights by using the L-BFGS algorithm.
AIWNB [20]	Filter-based attribute and instance-weighting naive Bayes, combining correlation-based attribute weights with frequency-based instance weights using eager learning AIWNB^E and similarity-based instance weights using lazy learning AIWNB^L.
Deep-learning models	
ResNet [203]	Adapted residual networks using 2 or 3 residual blocks.
FTT [8]	Adapted transformer with feature tokenizer.
PWe-dRVFL [204]	Combination of pruning-based and weighting-based ensemble deep random vector functional link neural network with re-normalization.

from real-world problems. The number of instances is distributed between 106 and 10992 and the number of attributes is distributed between 2 and 90. Some datasets contain missing values which are replaced by the mean or mode of the corresponding attribute. Besides, there are both nominal attributes and numerical attributes in some datasets. These 45 datasets provide a comprehensive evaluation of the proposed methods. The statistics of these datasets are summarized in Table 6.2. Similarly as in [9, 20, 22, 45], the classification accuracy of each method on each dataset is derived via stratified 10-fold cross-validation. For the proposed method, only 8 out of 9 folds of training data are used in training while the remaining one fold serves as validation data.

5.4.2 Comparisons to state-of-the-art discretization methods

The proposed discretization is compared with state-of-the-art discretization methods, Ameva [113], CAIM [23], MDLP [44], Modified Chi2 [102], PKID [84] and EMD [45] on the 45 datasets. The results are summarized in Table 5.3, where the results of the compared methods are obtained by using the KEEL tool [195]. The highest classification accuracy on each dataset among all compared methods is highlighted in bold. The

Table 5.2: Statistics of the benchmark datasets, where Inst., Attr., Class, Num., Nom. and Missing denote the number of instances, attributes, classes, numerical attributes, nominal attributes and whether the dataset contains missing values, respectively.

	Inst.	Attr.	Class	Num.	Nom.	Missing		Inst.	Attr.	Class	Num.	Nom.	Missing
abalone	4174	8	28	7	1	N	movement	360	90	15	90	0	N
appendicitis	106	7	2	7	0	N	newthyroid	215	5	3	5	0	N
australian	690	14	2	8	6	N	pageblocks	5472	10	5	10	0	N
auto	205	25	6	15	10	Y	penbased	10992	16	10	16	0	N
balance	625	4	3	4	0	N	phoneme	5404	5	2	5	0	N
banana	5300	2	2	2	0	N	pima	768	8	2	8	0	N
bands	539	19	2	19	0	Y	saheart	462	9	2	8	1	N
banknote	1372	5	2	5	0	N	satimage	6435	36	7	36	0	N
bupa	345	6	2	6	0	N	segment	2310	19	7	19	0	N
clevland	303	13	5	13	0	Y	seismic	2584	19	2	15	4	N
climate	540	18	2	18	0	N	sonar	208	60	2	60	0	N
contraceptive	1473	9	3	9	0	N	spambase	4597	57	2	57	0	N
crx	690	15	2	6	9	Y	specfheart	267	44	2	44	0	N
dermatology	366	34	6	34	0	Y	tae	151	5	3	5	0	N
ecoli	336	7	8	7	0	N	thoracic	470	17	2	3	14	N
flare-solar	1066	9	2	9	0	N	titanic	2201	3	2	3	0	N
glass	214	9	7	9	0	N	transfusion	748	5	2	5	0	N
haberman	306	3	2	3	0	N	vehicle	846	18	4	18	0	N
hayes	160	4	3	4	0	N	vowel	990	13	11	13	0	N
heart	270	13	2	13	0	N	wine	178	13	3	13	0	N
hepatitis	155	19	2	19	0	Y	wisconsin	699	9	2	9	0	N
iris	150	4	3	4	0	N	yeast	1484	8	10	8	0	N
mammographic	961	5	2	5	0	N							

average classification accuracy over all datasets is summarized at the bottom of Table 5.3.

As shown in Table 5.3, the naive Bayes classifier using the proposed MRmD discretization scheme obtains the highest classification accuracy on 24 datasets. Compared with the previous filter-based approaches, Ameva [113], CAIM [23], MDLP [44], Modified Chi2 [102] and PKID [84], the proposed MRmD discretization obtains an average improvement of 4.23%, 2.82%, 4.22%, 2.32% and 2.38%, respectively. As a filter-based method, the proposed MRmD outperforms the previously best discretization method, the wrapper-based algorithm, EMD [45], with an average improvement of 1.69% over the 45 datasets. The performance improvements on some datasets are significant. For example, the classification results of the proposed MRmD on “auto”, “bands” and “movement” are more than 8% higher than EMD [45]. Both “auto” and “movement” have a relatively small number of samples but a relatively large number of attributes. The NB classifier easily overfits to these two datasets. By maximizing the discriminant information and the generalization performance at the same time, the proposed MRmD achieves a much better generalization performance than EMD [45] that greedily maximizes the discriminant power for a small number of training samples.

Table 5.3: Comparisons of different discretization methods under the naive Bayes classification framework. The proposed MRmD achieves the best average classification accuracy, and outperforms the second best method, EMD [45], by 1.69% on average.

	Ameva [113]	CAIM [23]	MDLP [44]	Modified Chi2 [102]	PKID [84]	EMD [45]	MRmD
abalone	21.27±3.12	25.85±1.52	24.96±1.72	24.29±1.88	**26.11±2.25**	25.78±2.39	25.54±1.70
appendicitis	**88.00±9.58**	87.09±10.61	87.09±9.70	85.18±10.71	86.09±9.89	87.09±10.06	87.91±12.10
australian	85.07±3.99	**86.38±4.64**	84.49±4.32	84.49±4.16	85.51±3.68	85.65±3.46	86.37±3.86
autos	67.29±11.46	64.97±8.73	67.32±11.80	64.88±11.86	72.69±11.12	66.06±6.71	**76.93±10.75**
balance	79.68±4.07	80.31±4.23	72.66±6.53	90.88±1.50	**91.20±1.33**	85.44±3.81	91.04±1.70
banana	70.49±2.34	60.49±2.72	72.47±2.22	63.00±1.99	71.47±2.06	**73.57±1.99**	72.96±2.44
bands	72.55±4.31	66.43±3.94	50.45±9.07	72.35±6.18	68.65±7.04	65.50±5.52	**73.64±6.42**
banknote	89.43±2.24	89.07±2.31	92.05±1.55	91.40±2.33	92.20±2.27	**94.24±1.43**	91.55±1.64
bupa	65.99±12.29	61.69±8.91	57.15±7.40	63.76±4.60	62.77±9.86	65.48±8.94	**68.42±7.29**
cleveland	57.78±4.53	56.12±7.58	55.45±4.07	54.82±7.24	55.44±7.72	57.09±7.17	**58.07±8.89**
climate	91.11±1.61	91.67±1.71	93.52±2.52	91.30±2.87	90.19±2.49	**93.52±2.38**	93.51±3.38
contraceptive	50.64±4.73	49.63±2.89	50.51±4.84	50.24±3.15	50.92±2.83	52.00±3.44	**52.21±3.02**
crx	85.51±4.78	86.09±4.28	85.65±4.80	84.20±3.09	85.22±3.60	85.36±5.89	**86.23±6.25**
dermatology	98.10±2.23	97.55±2.99	97.82±2.14	**98.65±1.91**	97.82±2.51	94.82±3.30	98.63±1.95
ecoli	81.27±7.06	80.94±4.68	82.16±6.16	79.48±6.42	80.38±6.02	78.89±5.63	**84.28±5.05**
flare	65.57±4.94	65.57±4.94	67.54±3.80	65.29±4.94	65.48±4.93	67.26±4.09	**68.29±5.43**
glass	46.47±6.61	70.34±14.11	72.06±8.38	71.99±8.24	72.46±9.83	71.24±12.34	**75.67±9.03**
haberman	74.78±6.74	73.52±4.56	72.85±3.70	72.20±4.51	72.82±5.45	74.49±5.85	**74.80±4.29**
hayes	74.37±9.73	74.37±9.73	52.02±8.12	79.37±14.29	79.32±12.68	**81.57±11.15**	80.09±11.34
heart	83.70±8.59	84.07±7.21	84.07±8.91	82.96±9.43	**84.44±7.16**	82.96±8.31	84.07±8.91
hepatitis	81.96±9.64	83.88±10.37	83.83±11.62	82.63±12.40	80.71±11.69	82.54±10.21	**86.54±11.19**
iris	93.33±4.44	94.00±4.92	92.67±3.78	93.33±4.44	92.00±4.22	**95.33±4.27**	94.00±7.34
mammographic	81.48±4.38	82.62±4.04	82.21±4.46	82.63±5.19	83.25±5.48	**83.57±5.34**	83.36±4.16
movement	65.00±6.57	65.28±5.75	60.56±5.68	62.50±7.99	66.67±4.90	55.83±8.91	**68.90±8.84**
newthyroid	95.35±3.18	95.82±4.16	94.89±4.13	95.82±4.68	96.75±3.17	94.94±4.33	**97.66±3.96**
pageblocks	94.01±0.77	93.42±0.65	93.11±0.94	93.75±0.83	91.58±0.96	94.06±0.87	**94.10±0.85**
penbased	86.08±0.91	87.12±0.77	87.66±0.97	87.71±0.87	87.22±0.84	87.08±0.88	**88.67±0.86**
phoneme	78.89±1.95	78.94±1.82	76.89±2.14	77.05±1.33	77.42±2.18	**79.35±2.32**	79.13±2.30
pima	72.80±4.34	73.20±6.04	75.26±3.77	73.97±4.70	74.10±5.04	**77.22±3.40**	74.61±3.58
saheart	65.82±5.54	70.35±4.80	66.24±5.78	67.77±7.94	67.56±5.78	70.79±3.25	**70.79±3.58**
satimage	25.28±0.65	81.69±1.60	82.10±1.31	82.22±1.48	82.11±1.42	81.99±1.56	**82.28±1.37**
segment	91.26±1.08	90.39±1.19	91.04±1.59	89.87±2.20	89.09±2.74	**93.55±1.26**	92.29±1.69
seismic	82.24±2.76	81.96±2.87	82.00±2.24	85.80±1.62	82.47±1.68	93.34±0.24	**93.42±0.01**
sonar	77.88±9.10	77.45±8.30	76.88±12.68	78.36±8.25	74.52±14.03	73.93±10.33	**78.40±10.08**
spambase	89.95±1.60	89.38±1.18	89.89±1.40	90.21±1.43	89.45±1.59	**92.28±1.52**	90.53±1.75
specfheart	76.44±8.65	76.82±9.62	73.05±8.95	74.96±9.17	77.52±8.40	**81.28±3.72**	79.71±6.80
tae	51.13±15.38	49.04±17.35	34.42±2.36	55.71±10.84	49.04±17.63	54.38±10.92	**56.57±15.45**
thoracic	81.91±3.95	82.13±2.72	82.13±3.04	82.98±3.81	80.43±5.28	82.13±3.04	**82.98±3.47**
titanic	78.10±3.02	77.83±2.97	77.60±3.22	77.88±3.02	77.88±3.02	**78.33±3.07**	78.19±2.34
transfusion	76.87±6.65	76.33±2.20	75.00±4.79	75.00±2.98	74.99±5.28	74.06±4.34	**77.94±3.55**
vehicle	61.22±4.64	60.64±3.67	59.10±3.50	62.41±3.61	62.05±2.76	**64.19±4.62**	63.37±5.10
vowel	63.64±3.43	62.22±4.86	60.30±5.13	**65.15±4.32**	57.88±3.40	63.43±4.51	64.04±4.62
wine	98.30±2.74	97.75±3.92	**98.86±2.41**	95.98±7.79	96.63±4.70	92.12±7.24	98.30±2.74
wisconsin	96.71±1.91	96.71±1.91	97.28±2.18	97.14±2.13	97.28±2.18	95.13±2.16	**97.36±2.37**
yeast	56.95±3.05	57.96±4.19	56.95±3.25	56.20±3.97	55.19±3.87	57.35±3.96	**58.83±3.50**
AVG	74.93	76.34	74.94	76.84	76.78	77.47	**79.16**

Table 5.4: Ranks of the Wilcoxon test when comparing various discretization methods under naive Bayes classification framework. Large rank values in the first row and small rank values in the first column indicate that the proposed MRmD significantly outperforms all the compared discretization methods.

Algorithm	**MRmD**	**Ameva**	**CAIM**	**MDLP**	**Modified Chi2**	**PKID**	**EMD**
MRmD	-	1030.5	1024.5	1007.5	1014.0	1009.0	789.0
Ameva	4.5	-	546.5	606.5	499.0	545.0	348.0
CAIM	10.5	488.5	-	654.0	485.0	522.0	337.5
MDLP	27.5	428.5	381.0	-	456.0	470.5	303.5
Modified Chi2	21.0	536.0	550.0	579.0	-	522.5	358.5
PKID	26.0	490.0	513.0	564.5	512.5	-	341.0
EMD	246.0	687.0	697.5	731.5	676.5	694.0	-

To evaluate the significance of the performance gains, we apply the Wilcoxon signed-rank test [205] to thoroughly compare each pair of algorithms. The Wilcoxon signed-rank test is a non-parametric statistical test, which ranks the performance of any two algorithms for each dataset, and compares the ranks for their differences. Table 5.4 presents the detailed ranks computed by the Wilcoxon test using naive Bayes classifier. Each entry $R_{i,j}$ in Table 5.4 is the sum of ranks for all datasets on which the algorithm in the i-th row is compared with the algorithm in the j-th column. For the confidence level of $\alpha = 0.05$ and $N = 45$, $R_{i,j} > 692$ indicates that the algorithm in the i-th row is significantly better than the algorithm in the j-th column. As shown in Table 5.4, the proposed MRmD discretization significantly better than Ameva ($R_{1,2} = 1030.5$), CAIM ($R_{1,3} = 1024.5$), MDLP ($R_{1,4} = 1007.5$), Modified Chi2 ($R_{1,5} = 1014$), PKID ($R_{1,6} = 1009$) and EMD ($R_{1,7} = 789$). These results clearly demonstrate that the proposed MRmD significantly outperforms all the compared discretization methods.

The proposed MRmD discretization method can be used to boost the performance of not only naive Bayes, but also many other classifiers such as decision tree. We hence include decision tree (C4.5) [90] in the comparison, and summarize the results in Table 5.5. The proposed MRmD discretizer obtains the highest classification accuracy on 27 datasets. Compared with the previous filter-based methods, Ameva [113], CAIM [23], MDLP [44], Modified Chi2 [102] and PKID [84], the proposed MRmD discretization achieves an average improvement of 4.82%, 3.73%, 3.97%, 4.41% and 7.47%, respectively. Compared with the previous best discretization method, EMD [45], the proposed MRmD achieves an average improvement of 1.45% over the 45 datasets. Similarly, we conduct the Wilcoxon signed-rank test [205] on each pair of algorithms to evaluate the significance of the performance gains. As shown in Table 5.6, the proposed MRmD discretization using C4.5 significantly better than Ameva ($R_{1,2} = 955.5$), CAIM ($R_{1,3} = 970$), MDLP ($R_{1,4} = 998$), Modified Chi2 ($R_{1,5} = 1014$), PKID ($R_{1,6} = 1024.5$) and EMD ($R_{1,7} = 858$).

Table 5.5: Comparisons of different discretization methods under the C4.5 classification framework. The proposed MRmD achieves the best average classification accuracy, and outperforms the second-best method, EMD [45], by 1.45% on average.

	Ameva [113]	CAIM [23]	MDLP [44]	Modified Chi2 [102]	PKID [84]	EMD [45]	MRmD
abalone	21.49±2.13	24.32±1.79	25.37±2.25	17.49±2.55	23.07±2.16	23.43±2.15	**25.63±1.66**
appendicitis	83.36±12.55	83.36±12.55	83.36±10.99	78.55±12.64	80.18±2.77	87.00±7.15	**94.36±6.43**
australian	86.67±3.47	87.25±4.09	86.38±3.36	85.80±4.20	84.93±3.36	85.36±4.27	**88.55±4.32**
autos	75.49±5.66	72.63±8.64	76.91±10.29	**78.97±9.92**	76.70±10.52	76.44±7.39	78.52±6.29
balance	74.56±4.55	74.72±4.58	69.92±5.52	66.40±5.75	64.82±5.35	**80.47±3.95**	77.13±3.58
banana	72.49±2.06	63.87±1.58	74.85±2.20	63.92±1.80	70.43±1.84	87.30±1.55	**87.66±1.25**
bands	67.00±5.62	64.58±4.89	53.78±9.16	66.42±4.23	61.97±7.01	64.94±6.61	**74.58±4.79**
banknote	90.96±2.09	88.92±2.04	94.46±1.88	95.26±1.81	84.84±2.33	96.57±1.46	**98.32±1.18**
bupa	68.07±5.37	60.65±9.17	57.15±7.40	57.04±6.45	57.89±3.33	68.14±7.84	**73.03±3.69**
cleveland	55.74±7.41	54.84±7.07	53.77±5.33	54.71±9.70	53.45±5.35	54.80±6.52	**57.12±4.16**
climate	92.78±2.68	92.78±2.55	93.33±2.64	91.67±1.49	91.48±0.91	93.33±2.06	**95.74±2.49**
contraceptive	49.09±3.62	51.05±3.16	50.45±2.69	50.45±4.73	48.75±2.99	52.61±3.70	**53.98±3.10**
crx	85.51±5.20	87.39±3.87	86.81±3.89	**87.68±4.70**	85.22±4.92	86.38±3.44	86.81±3.58
dermatology	95.35±3.86	93.18±5.58	95.89±2.97	95.89±2.97	94.54±5.76	94.82±3.30	**96.45±3.00**
ecoli	70.82±5.58	74.69±4.94	77.71±5.86	73.81±4.14	66.03±6.90	74.72±7.72	**79.17±6.71**
flare	67.82±4.28	67.82±4.28	67.54±3.80	67.54±3.80	67.54±3.80	67.26±4.09	**67.92±4.61**
glass	53.05±5.13	67.61±11.90	**75.79±10.53**	62.80±13.03	57.88±11.59	73.70±9.88	74.24±7.77
haberman	74.13±6.07	**75.12±6.01**	72.53±3.38	73.53±1.00	73.53±1.00	74.15±3.65	74.81±4.54
hayes	80.20±7.16	**80.20±7.16**	52.02±8.12	72.01±13.25	71.07±12.98	74.26±9.54	71.88±7.53
heart	78.89±9.57	78.52±11.29	79.63±9.44	78.89±9.57	79.26±7.45	**82.22±7.18**	80.37±4.98
hepatitis	78.21±11.80	82.71±8.27	83.25±7.63	80.08±8.97	82.58±6.82	80.71±6.95	**83.92±6.40**
iris	93.33±3.14	93.33±3.14	93.33±3.14	93.33±4.44	92.67±6.63	**95.33±4.27**	94.67±7.77
mammographic	81.59±4.78	82.84±5.10	**83.15±5.29**	82.00±4.41	81.17±5.25	83.15±5.36	82.83±2.76
movement	43.33±8.30	47.50±7.69	60.56±10.29	63.06±8.18	32.22±10.33	63.61±5.19	**64.17±9.42**
newthyroid	92.53±5.08	93.51±5.02	94.44±4.21	93.98±4.43	93.98±3.09	**94.94±4.33**	94.87±4.90
pageblocks	96.56±0.52	96.18±0.50	96.84±0.60	94.74±1.07	94.96±0.66	96.93±0.65	**97.17±0.50**
penbased	92.93±0.58	88.77±1.35	88.66±1.29	89.04±1.19	67.00±0.61	**94.91±0.64**	94.76±0.73
phoneme	78.77±1.96	79.13±1.81	81.24±2.25	75.33±1.40	76.81±1.84	**84.47±1.74**	83.85±1.59
pima	**74.48±3.27**	73.45±5.28	73.44±4.25	72.03±4.68	73.07±5.89	74.35±1.75	74.36±3.24
saheart	69.91±4.99	70.55±3.94	68.17±5.55	69.71±4.62	65.80±1.36	68.39±5.13	**71.66±4.27**
satimage	25.07±0.55	85.41±1.30	84.54±1.55	83.87±1.39	80.20±0.91	84.83±1.25	**86.01±1.32**
segment	95.37±1.12	94.68±1.31	93.85±1.45	88.31±2.23	84.76±1.79	96.06±0.76	**96.06±1.22**
seismic	93.42±0.01	93.42±0.01	93.42±0.01	93.42±0.01	93.42±0.01	93.34±0.24	**93.42±0.01**
sonar	**79.69±11.88**	74.00±6.63	76.38±11.94	73.98±11.05	69.62±10.68	77.31±9.65	77.86±3.96
spambase	93.54±1.20	**93.56±1.28**	92.73±1.30	87.64±1.62	88.69±1.38	92.52±1.13	92.89±1.07
specfheart	80.50±6.80	77.85±6.91	72.68±9.79	77.52±6.67	79.42±1.75	82.04±2.11	**82.05±4.55**
tae	44.54±19.82	45.83±16.39	34.42±2.36	52.96±12.40	47.08±12.93	53.17±12.30	**58.17±12.88**
thoracic	84.68±0.85	84.68±0.85	84.68±0.85	85.11±0.00	85.11±0.00	84.68±0.85	**85.11±0.00**
titanic	77.33±3.04	77.74±3.15	77.15±2.90	77.60±2.96	78.92±2.31	**79.06±2.21**	77.60±2.42
transfusion	79.27±4.67	77.27±2.48	76.21±0.41	76.21±0.41	76.21±0.41	77.28±4.54	**79.96±4.00**
vehicle	67.73±4.17	67.39±4.15	68.32±5.09	68.31±5.59	64.54±4.47	68.31±3.67	**70.10±2.33**
vowel	72.83±5.68	69.60±3.35	**73.23±6.43**	71.21±5.52	48.48±4.29	70.71±5.03	72.72±4.78
wine	**93.82±4.86**	91.01±5.53	89.84±7.98	92.68±6.48	79.74±11.09	92.12±7.24	93.79±5.98
wisconsin	93.71±2.04	93.85±1.92	94.42±2.81	94.71±3.09	93.84±2.72	94.99±1.48	**95.60±3.09**
yeast	54.99±3.30	53.04±4.20	57.22±3.16	44.34±3.06	38.62±3.41	52.43±3.49	**58.89±4.08**
AVG	75.15	76.24	76.00	75.56	72.50	78.52	**79.97**

Table 5.6: Ranks of the Wilcoxon test when comparing discretization methods using C4.5.

Algorithm	**MRmD**	**Ameva**	**CAIM**	**MDLP**	**Modified Chi2**	**PKID**	**EMD**
MRmD	-	955.5	970.0	998.0	1014.0	1024.5	858.0
Ameva	79.5	-	555.0	508.5	640.5	825.5	290.5
CAIM	65.0	480.0	-	500.5	665.5	890.0	226.5
MDLP	37.0	526.5	534.5	-	635.5	831.0	266.5
Modified Chi2	21.0	394.5	369.5	399.5	-	771.5	157.5
PKID	10.5	209.5	145.0	204.0	263.5	-	33.0
EMD	177.0	744.5	808.5	768.5	877.5	1002.0	-

Table 5.7: Comparisons with the state-of-the-art classifiers. The proposed MRmD-RNB significantly outperforms the previous best naive Bayes method, RNB [9], by 2.84% on average. Compared with the best deep-learning method, FTT [8], the proposed MRmD-RNB obtains an improvement of 0.93% on average.

	WANBIA [21]	CAWNB [22]	AIWNB^L [20]	AIWNB^E [20]	RNB [9]	ResNet [203]	PWedRVFL [204]	FTT [8]	MRmD-RNB
abalone	26.71±1.51	25.15±1.85	26.09±1.44	24.01±1.30	26.78±1.63	25.01±6.25	26.35±1.40	26.98±2.09	**27.00±1.90**
appendicitis	87.55±9.07	87.55±9.07	84.91±9.07	84.91±9.07	87.55±9.07	86.91±8.85	82.91±10.47	85.82±6.27	**88.64±8.49**
australian	86.81±3.19	86.81±4.64	85.35±4.61	84.77±4.65	86.80±4.34	67.32±5.95	79.99±4.82	86.66±2.94	**86.94±5.14**
auto	75.62±5.41	76.15±11.05	71.26±8.47	71.31±9.16	81.13±9.13	65.38±9.95	30.57±10.00	75.78±7.15	**84.35±9.31**
balance	71.86±3.89	71.86±3.89	70.08±2.91	71.53±3.14	71.86±3.89	88.79±3.93	87.68±1.24	89.93±4.48	**91.04±1.70**
banana	72.83±2.31	73.38±2.04	73.32±2.00	71.98±2.49	73.38±2.04	73.72±4.06	87.25±2.13	**87.42±1.79**	73.36±1.88
bands	70.49±5.86	70.69±6.84	70.12±6.23	70.12±6.23	70.69±6.84	66.60±7.04	64.57±4.35	69.01±6.72	**75.31±3.89**
banknote	92.13±1.40	92.78±1.83	92.57±1.48	92.06±1.38	92.78±1.83	91.70±17.28	**99.93±0.22**	93.81±2.26	92.86±1.87
bupa	53.27±10.03	53.27±10.03	42.02±0.89	42.02±0.89	53.27±10.03	70.09±9.85	68.13±8.41	**70.73±6.81**	70.49±8.43
clevland	57.73±5.62	58.45±4.74	58.15±4.65	57.17±5.59	58.57±7.37	58.40±4.31	58.76±5.91	56.55±6.37	**59.06±6.35**
climate	94.26±2.66	94.26±2.66	94.26±2.66	94.26±2.66	94.26±2.66	91.48±0.96	91.13±2.23	93.87±2.65	**94.45±4.16**
contraceptive	51.38±4.72	51.79±4.33	51.12±4.16	50.72±3.98	52.34±4.58	52.01±4.82	52.21±4.61	**53.57±1.78**	53.43±2.73
crx	**87.11±5.59**	86.67±5.29	85.94±5.17	85.21±5.26	86.38±4.97	78.41±11.13	80.73±4.47	86.67±4.74	86.82±5.20
dermatology	98.64±1.92	98.37±2.28	97.56±2.00	97.56±2.00	98.64±1.92	97.27±3.62	96.17±3.88	97.82±2.67	**98.65±2.30**
ecoli	82.51±4.00	83.38±3.58	82.23±5.18	82.23±5.18	83.39±3.53	83.96±6.34	84.03±5.22	83.03±5.54	**84.59±5.59**
flare-solar	68.01±4.74	68.01±4.74	68.20±4.99	68.20±4.99	68.20±4.99	65.87±3.73	67.64±4.37	67.82±5.64	**68.29±5.43**
glass	71.13±8.74	72.47±5.30	75.28±8.17	74.28±7.23	71.97±4.72	56.10±9.21	64.61±7.21	70.08±8.97	**75.71±8.38**
haberman	73.18±3.90	73.18±3.90	26.47±0.72	26.47±0.72	73.18±3.90	73.21±5.01	72.81±8.46	68.94±6.89	**74.80±4.29**
hayes	60.03±1.42	60.03±1.42	60.03±1.42	60.03±1.42	60.03±1.42	77.50±15.65	66.28±15.60	78.17±9.62	**80.09±11.34**
heart	85.19±8.90	85.56±8.27	83.70±9.91	83.70±9.91	85.19±8.90	81.85±8.09	78.89±11.60	81.85±10.66	**85.93±9.04**
hepatitis	82.17±11.98	83.42±9.34	82.13±13.16	82.79±12.83	84.04±9.83	85.75±8.51	83.42±8.40	84.63±10.94	**87.25±8.85**
iris	93.33±6.29	93.33±6.29	92.67±6.63	92.67±6.63	93.33±6.29	95.33±6.33	**96.00±3.27**	96.00±4.42	94.00±4.92
mammographic	82.52±4.20	82.52±4.35	82.32±4.00	82.42±3.83	82.63±4.38	81.27±4.04	81.29±3.98	82.31±3.74	**83.36±4.94**
movement	67.16±3.63	68.45±6.00	67.12±4.56	64.63±5.09	68.75±8.39	71.95±9.75	**75.15±7.47**	71.14±6.84	71.81±8.97
newthyroid	95.80±3.50	95.35±3.18	95.32±4.98	95.76±4.73	95.80±3.50	94.91±4.56	89.35±3.51	95.78±5.40	**98.57±3.21**
pageblocks	96.04±0.87	96.40±0.70	93.79±1.13	93.02±1.37	96.35±0.97	93.48±1.65	94.72±0.62	96.24±0.77	**96.46±0.78**
penbased	89.88±0.68	92.91±0.75	93.60±0.67	88.82±0.95	93.12±0.72	94.08±1.72	**95.84±1.35**	94.00±0.73	93.70±0.69
phoneme	80.27±1.74	80.14±1.87	79.79±1.38	78.13±1.36	79.94±1.76	79.29±2.74	82.59±1.78	**82.68±1.36**	80.50±2.25
pima	74.21±5.02	75.12±5.88	73.82±4.90	73.69±4.58	74.86±5.70	74.47±2.92	75.24±5.51	74.61±5.04	**75.90±3.56**
saheart	70.35±3.06	69.91±3.93	67.74±5.10	67.74±5.10	70.12±4.70	71.44±5.89	72.97±4.16	68.83±5.62	**73.38±4.84**
satimage	84.40±1.08	84.27±1.22	85.44±0.87	81.40±1.43	85.86±0.86	82.18±1.41	85.16±1.28	84.99±0.54	**86.12±1.14**
segment	94.72±1.22	93.77±1.23	94.20±1.76	92.64±1.81	94.50±1.02	92.77±2.87	90.22±1.51	**95.11±1.57**	94.59±1.12
seismic	93.42±0.01	93.42±0.01	82.66±3.04	81.19±2.97	93.42±0.01	93.38±0.13	89.86±1.77	93.03±0.81	**93.46±0.12**
sonar	78.37±9.17	76.99±10.02	76.97±9.42	76.49±9.84	77.42±9.46	77.93±10.01	76.53±6.63	**81.14±9.98**	80.28±10.73
spambase	93.78±0.97	94.07±0.75	90.16±1.33	89.94±1.32	93.98±1.29	91.11±1.19	89.83±1.93	93.57±1.01	**94.46±0.69**
specfheart	78.54±8.72	78.56±7.28	75.07±11.65	75.07±11.65	81.15±9.48	78.26±5.54	76.36±5.48	78.21±6.00	**82.70±7.77**
tae	34.40±1.79	34.40±1.79	32.44±1.61	32.44±1.61	34.40±1.79	47.13±9.51	47.60±15.25	52.92±12.41	**58.57±15.67**
thoracic	83.83±1.49	83.40±1.96	82.34±2.25	81.91±3.05	83.83±1.79	84.90±0.67	**85.53±0.85**	84.26±1.70	84.26±1.79
titanic	77.60±2.40	77.60±2.40	77.60±2.40	77.60±2.40	77.60±2.40	77.88±1.48	**79.10±1.00**	79.05±1.53	77.74±2.43
transfusion	76.21±0.43	76.21±0.43	74.47±4.40	74.47±4.40	76.21±0.43	76.21±0.43	77.80±4.58	**79.68±2.37**	77.80±3.91
vehicle	65.62±5.28	65.61±3.74	64.66±4.05	61.36±6.21	67.73±3.21	69.03±5.88	**78.16±6.32**	74.70±2.99	69.87±5.40
vowel	64.14±5.50	64.55±4.72	66.87±4.99	63.64±4.69	64.65±5.28	64.15±3.76	**71.52±5.47**	68.69±4.52	65.15±5.50
wine	98.30±2.74	97.19±3.96	96.60±2.93	97.71±2.96	98.30±2.74	93.33±7.31	96.01±3.71	96.63±2.75	**98.30±2.74**
wisconsin	96.93±2.61	97.22±2.32	97.07±2.75	97.36±2.37	97.22±2.78	96.64±2.07	95.61±2.27	95.47±2.71	**97.36±2.16**
yeast	56.75±4.29	57.42±4.32	57.15±4.00	57.15±3.64	57.35±4.19	55.59±4.70	55.47±3.02	56.74±3.22	**59.30±3.06**
AVG	77.23	77.38	75.13	74.50	77.75	77.20	77.38	79.66	**80.59**

Table 5.8: Ranks of the Wilcoxon test when comparing state-of-the-art classifiers.

Algorithm	**MRmD-RNB**	**WANBIA**	**CAWNB**	**AIWNB^L**	**AIWNB^E**	**RNB**	**ResNet**	**PWedRVFL**	**FTT**
MRmD-RNB	-	1019.5	1034.0	1015.0	1034.0	1029.5	998.5	788.0	751.0
WANBIA	15.5	-	408.0	817.0	946.0	232.0	593.5	496.0	334.0
CAWNB	1.0	627.0	-	868.0	973.0	262.5	621.5	507.0	358.0
AIWNB^L	20.0	218.0	167.0	-	780.5	94.0	450.0	376.0	145.0
AIWNB^E	1.0	89.0	62.0	254.5	-	42.5	341.0	337.5	115.5
RNB	5.5	803.0	772.5	941.0	992.5	-	676.0	543.0	433.5
ResNet	36.5	441.5	413.5	585.0	694.0	359.0	-	448.0	182.0
PWedRVFL	247.0	539.0	528.0	659.0	697.5	492.0	587.0	-	284.0
FTT	284.0	701.0	677.0	890.0	919.5	591.5	853.0	751.0	-

5.4.3 Comparisons to state-of-the-art classifiers

The proposed MRmD is integrated with RNB [9], named MRmD-RNB, and compared with five state-of-the-art NB classifiers including WANBIA [21], CAWNB [22], AIWNB^L [20], AIWNB^E [20] and RNB [9], and three deep-learning models including ResNet [203], FTT [8] and PWedRVFL [204].

As shown in Table 5.7, the proposed MRmD-RNB obtains the highest classification accuracy on 29 datasets out of 45 datasets among all the compared methods. Compared with the recent attribute weighting methods, AIWNB^L and AIWNB^E [20], the proposed MRmD-RNB achieves an average improvement of 5.46% and 6.09%, respectively. Compared with the previous best naive Bayes classifier, RNB [9], the proposed MRmD-RNB achieves an average improvement of 2.84%. The performance gains on some of the datasets are significant. For example, the performance gains on "balance", "bupa", "hayes" and "tae" are more than 17% over RNB [9]. Among them, both "hayes" and "tae" have a relatively small number of instances and attributes. The NB classifier may overfit to these small datasets due to few training samples. The proposed discretization method greatly enhances the generalization ability of the state-of-the-art NB classifier and hence significantly improves the classification performance.

The proposed MRmD-RNB achieves a higher average classification accuracy than the three compared deep-learning models. We conjecture that due to the lack of sufficient representative training samples, the deep-learning models may overfit to the training data. The proposed MRmD-RNB well boosts the generalization capability in both discretization and classifier design, and hence demonstrates excellent classification performance.

We also conduct the Wilcoxon signed-rank test on the performance gains over the state-of-the-art classifiers. As shown in Table 5.8, the proposed MRmD-RNB significantly outperforms all the compared methods, as all the ranks in the first row are much larger than the significance value of 692.

5.4.4 Ablation study

To evaluate the performance gain brought by the generalization capability, we compare the proposed method with the following discretization methods under the naive Bayes classification framework.

MR^O: Only the Max-Relevance criterion is used. This serves as the baseline which does not consider the generalization capability at all.

MR^C: Only the Max-Relevance criterion is used, but the number of cut points is restricted to the number of classes, as in CAIM [23] and CACC [25].

MR^T: Only the Max-Relevance criterion is used, but the number of cut points is restricted to the twice number of cut points derived by MRmD. The average accuracy over 10-fold cross-validation and the average number of cut points are summarized in Table 5.9 and the classification accuracy across datasets for comparing MRmD with other three variants are presented in Fig. 5.2 where the classification accuracies over all datasets are sorted in ascending order with respect to MRmD. As shown in Fig. 5.2, the proposed MRmD often achieves the highest classification accuracy among the compared

Table 5.9: Comparisons of different discretization methods when constraining the number of cut points using: 1) the number of classes, **MR**^{C}; 2) MRmD criterion, **MRmD**; 3) twice the number of cut points derived by MRmD, **MR**^{T}; 4) no constraint at all, **MR**^{O}.

	MR^C		**MRmD**		**MR**^{T}		**MR**^{O}	
	Acc	# of Cut Points	Acc	# of Cut Points	Acc	# of Cut Points	Acc	# of Cut Points
abalone	25.32±1.64	198.00± 0.00	**25.54±1.70**	71.30±3.89	25.46±1.83	140.60±7.78	19.55±1.76	5515.80±12.36
appendicitis	85.00±7.82	14.00± 0.00	**87.91±12.10**	3.70±1.34	85.18±10.52	7.40±2.67	84.91±7.75	138.60±7.50
australian	86.22±3.89	24.00± 0.00	86.37±3.86	29.50±4.55	**87.09±3.62**	55.60±8.93	71.44±3.69	717.80±7.86
auto	65.83±8.00	123.80± 0.42	76.93±10.75	204.30±11.71	75.59±9.98	382.10±20.55	**78.46±10.87**	759.00±7.47
balance	86.40±2.01	12.00± 0.00	91.04±1.70	14.70±1.25	78.10±6.45	16.00±0.00	**91.84±0.48**	16.00±0.00
banana	58.77±2.02	4.00± 0.00	**72.96±2.44**	16.60±4.14	70.43±3.31	33.20±8.28	60.85±1.83	2529.70±11.58
bands	66.60±4.01	38.00± 0.00	73.64±6.42	34.10±1.85	69.94±6.90	67.20±4.13	**75.88±3.60**	643.90±5.99
banknote	85.13±4.33	8.00± 0.00	**91.55±1.64**	28.30±3.50	88.19±3.43	56.60±7.00	82.14±3.17	1671.60±14.19
bupa	60.27±7.49	12.00± 0.00	**68.42±7.29**	13.40±2.46	61.71±10.08	26.80±4.92	60.57±8.33	251.90±5.38
clevland	57.36±6.12	37.00± 0.00	58.07±8.89	17.70±2.83	**59.12±2.79**	3.80±2.04	53.24±5.80	319.60±3.06
climate	91.10±1.73	36.00± 0.00	**93.51±3.38**	12.00±1.83	91.66±1.83	24.00±3.65	91.49±0.86	1337.10±15.88
contraceptive	46.44±2.76	21.00± 0.00	**52.21±3.02**	12.40±1.26	46.58±2.87	19.80±2.04	50.65±2.95	61.70±0.48
crx	84.93±6.45	26.00± 0.00	**86.23±6.25**	12.80±0.92	84.93±6.45	22.90±1.79	74.77±4.15	721.80±5.83
dermatology	97.01±2.97	89.30± 0.67	**98.63±1.95**	40.40±0.97	95.03±4.11	70.00±1.33	96.99±3.03	140.30±1.06
ecoli	82.42±6.18	41.90± 0.32	**84.28±5.05**	15.50±1.35	79.98±7.44	29.10±2.77	68.81±8.30	288.90±2.23
flare-solar	68.01±5.45	12.00± 0.00	68.29±5.43	11.30±1.42	68.29±5.92	15.70±1.89	**68.66±6.14**	17.60±0.70
glass	68.15±6.77	54.00± 0.00	**75.67±9.03**	18.30±1.89	64.58±8.04	36.60±3.78	57.91±9.16	635.40±10.73
haberman	73.58±9.21	6.00± 0.00	**74.80±4.29**	2.90±1.20	72.92±6.66	5.80±2.39	71.59±4.32	75.70±2.50
hayes	**84.05±10.19**	11.00± 0.00	80.09±11.34	7.90±1.10	80.44±10.04	9.70±1.25	83.46±8.16	11.00±0.00
heart	81.11±9.79	22.50± 0.71	**84.07±8.91**	9.80±1.32	81.48±10.33	17.80±2.30	71.85±8.04	265.50±2.95
hepatitis	82.63±7.83	34.60± 0.97	**86.54±11.19**	39.50±8.63	83.25±7.96	73.90±16.97	81.42±7.81	185.20±3.85
iris	91.33±5.49	12.00± 0.00	**94.00±7.34**	10.60±1.43	90.00±7.86	21.20±2.86	92.00±8.20	52.80±2.04
mammographic	80.96±5.21	10.00± 0.00	**83.36±4.16**	19.70±4.81	82.62±3.17	34.60±8.83	82.42±4.14	75.90±1.52
movement	67.88±9.12	1350.00± 0.00	**68.90±8.84**	1395.90±43.86	68.02±6.06	2791.80±87.73	48.57±8.64	16060.10±147.14
newthyroid	95.82±4.05	15.00± 0.00	**97.66±3.96**	12.30±1.06	95.80±3.44	24.60±2.12	95.35±5.83	133.40±3.89
pageblocks	90.55±1.45	50.00± 0.00	**94.10±0.85**	454.50±31.84	93.15±1.28	909.00±63.68	93.75±1.00	3147.50±27.28
penbased	83.55±0.97	160.00± 0.00	**88.67±0.86**	252.30±5.95	86.28±0.84	504.60±11.89	87.85±0.88	1581.40±2.12
phoneme	76.26±2.25	10.00± 0.00	**79.13±2.30**	6.00±0.00	76.06±2.42	12.00±0.00	75.83±1.68	5589.50±19.03
pima	73.03±4.57	16.00± 0.00	**74.61±3.58**	64.00±11.24	72.91±4.52	128.00±22.49	67.57±6.11	783.60±9.09
saheart	69.91±6.18	17.00± 0.00	**70.79±3.58**	18.30±4.06	70.56±6.07	35.60±8.11	59.07±5.94	936.70±8.97
satimage	79.81±1.70	216.00± 0.00	82.28±1.37	559.70±9.39	82.11±1.42	1119.40±18.79	**82.39±1.23**	2245.80±5.20
segment	85.76±2.54	116.90± 0.32	**92.29±1.69**	205.50±8.14	89.48±2.05	406.70±16.47	83.03±1.38	9186.50±34.23
seismic	85.64±2.60	27.00± 0.00	**93.42±0.01**	0.40±0.70	93.42±0.01	0.80±1.40	91.14±1.67	1025.40±8.76
sonar	75.49±10.49	120.00± 0.00	**78.40±10.08**	43.90±4.28	74.99±10.69	87.80±8.56	63.82±12.68	4275.50±39.92
spambase	90.98±1.37	113.90± 0.32	90.53±1.75	341.20±31.30	90.55±1.44	660.30±59.18	**91.24±1.12**	7480.40±18.82
specfheart	75.96±8.83	88.00± 0.00	**79.71±6.80**	172.80±15.53	77.83±7.36	341.90±27.99	79.38±4.87	946.40±12.95
tae	46.11±12.83	11.00± 0.00	56.57±15.45	55.10±6.64	**58.58±16.50**	78.40±3.50	57.46±10.58	83.00±1.41
thoracic	81.91±2.70	22.00± 0.00	**82.98±3.47**	9.40±2.07	82.55±2.62	14.00±3.27	81.28±4.11	193.10±3.28
titanic	77.96±2.24	4.00± 0.00	**78.19±2.34**	2.90±0.32	77.96±2.24	3.90±0.32	77.87±2.35	5.00±0.00
transfusion	75.14±4.74	8.00± 0.00	**77.94±3.55**	18.50±3.54	76.33±3.02	37.00±7.07	73.53±4.46	139.40±4.17
vehicle	58.16±5.82	72.00± 0.00	**63.37±5.10**	178.60±13.23	61.10±4.61	353.20±23.86	61.58±4.80	1174.40±4.53
vowel	56.67±6.52	123.00± 0.00	**64.04±4.62**	85.40±5.02	54.75±5.34	169.70±9.31	23.03±3.77	6479.50±14.74
wine	94.87±6.85	39.00± 0.00	**98.30±2.74**	16.70±0.48	94.31±8.58	33.40±0.97	92.16±4.74	655.60±7.89
wisconsin	96.78±2.64	18.00± 0.00	**97.36±2.37**	18.00±1.25	96.63±2.40	36.00±2.49	97.36±2.37	70.50±1.27
yeast	45.85±6.05	63.00± 0.00	**58.83±3.50**	17.10±1.10	39.98±3.76	33.20±2.20	50.47±3.60	368.00±2.75
AVG	75.39	77.93	**79.16**	101.67	76.35	198.93	73.44	1755.41

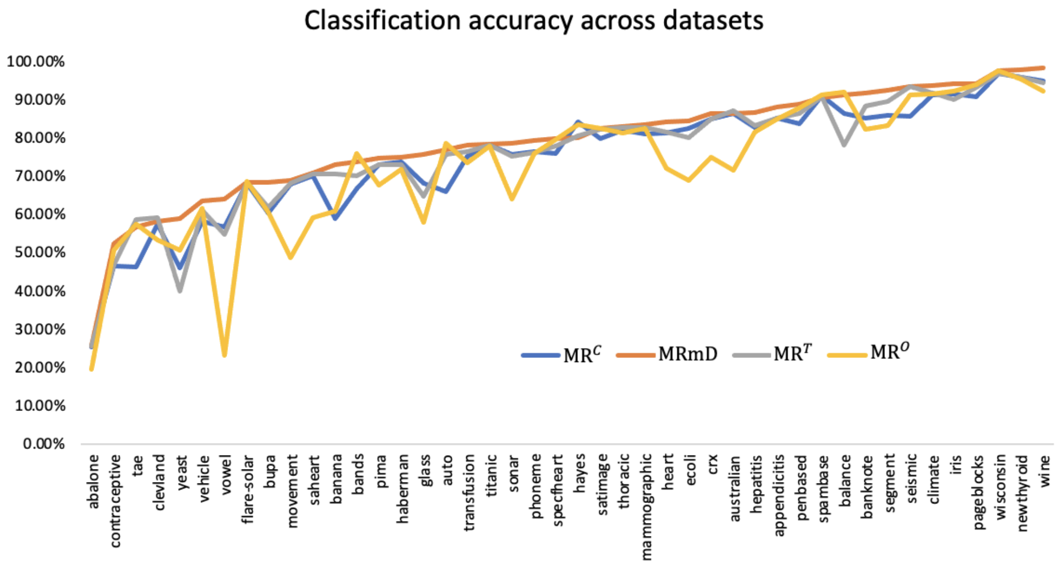


FIGURE 5.2: Classification accuracy across datasets for comparisons of the Max-Relevance criterion with different constrains.

methods. As detailed in Table 5.9, the proposed MRmD obtains the highest classification accuracy averaged over 45 datasets. Compared with $\mathbf{MR}^O$ that does not consider the generalization capability, the proposed MRmD achieves an improvement of 5.72% on average, which is the performance gain brought by the generalization capability through the proposed MRmD discretization scheme. By restricting the number of cut points to the number of classes, $\mathbf{MR}^C$ enhances the generalization capability, but it may lead to a severe loss in discriminant information, as the number of classes may be as small as 2. The proposed MRmD hence outperforms $\mathbf{MR}^C$ by 3.77% on average. $\mathbf{MR}^T$ utilizes twice as many cut points as MRmD, which leads to a decrease of 2.81% on average from MRmD. This set of results demonstrate that the proposed MRmD can better balance the discriminant power and generalization capability, thus achieving higher classification accuracy.

5.5 Summary

Previous data discretization methods often overemphasize maximizing the discriminant information while overlooking the primary goal of data discretization in classification, *i.e.*, to enhance the generalization ability of a classifier. To address this problem, a

Maximal-Dependency-Minimal-Divergence scheme is proposed to simultaneously maximize the generalization capability and discriminant information. The proposed MDmD criterion is difficult to implement in practice due to the difficulty in estimating the high-order mutual information. We hence propose a more practical solution, Maximal-Relevance-Minimal-Divergence criterion, which discretizes one attribute at a time in a top-down manner. The proposed MRmD criterion generates a discretization scheme with a trade-off between retaining the discriminant information and improving the generalization ability for the subsequent classifier. Experimental results on the 45 benchmark datasets demonstrate that the proposed MRmD significantly outperforms all the compared discretization methods and, by integrating the proposed MRmD with RNB, the resulting MRmD-RNB significantly outperforms all the compared classifiers.

The performance gain of the proposed MRmD may be limited by two factors: 1) The greedy top-down hierarchical splitting algorithm only leads to a near-optimal discretization scheme. Other more sophisticated search algorithms such as genetic algorithms and hyperheuristics can be used to approximate the optimal solution better. 2) The proposed MRmD simplifies the MDmD criterion by ignoring the high-order feature interaction as it is difficult to reliably estimate the multivariate distributions, which may lead to some information loss. A possible improvement is to consider the high-order feature interaction when designing the discretization criterion.

Chapter 6

Boosting the Discriminant Power of Naive Bayes via Feature Augmentation

Naive Bayes has been widely used in many applications because of its simplicity and ability in handling both numerical data and categorical data[1]. However, lack of modeling of correlations between features limits its performance. In addition, noise and outliers in the real-world dataset also greatly degrade the classification performance. In this paper, we propose a feature augmentation method employing a stack auto-encoder to reduce the noise in the data and boost the discriminant power of naive Bayes. The proposed stack auto-encoder consists of two auto-encoders for different purposes. The first encoder shrinks the initial features to derive a compact feature representation in order to remove the noise and redundant information. The second encoder boosts the discriminant power of the features by expanding them into a higher-dimensional space so that different classes of samples could be better separated in the higher-dimensional space. By integrating the proposed feature augmentation method with the regularized naive Bayes, the discrimination power of the model is greatly enhanced. The proposed method is evaluated on a set of machine-learning benchmark datasets. The experimental

[1] This work has been published in the 2021 International Conference on Pattern Recognition [184] and the extended version has been submitted to Pattern Recognition.

results show that the proposed method significantly and consistently outperforms the state-of-the-art naive Bayes classifiers.

6.1 Introduction

Naive Bayes (NB) has been widely used in many applications, e.g., text classification [17, 206, 207], action recognition [208], scene recognition [209] and malware detection [210]. Naive Bayes is a simple and effective classification model. One notable advantage of NB is its ability of handling mixed data types, e.g., both categorical and numerical data. For simplicity, it often assumes that features are independent to each other conditioned on the classification variable. However, the independence assumption rarely holds in reality.

To address this problem, many approaches have been developed, e.g., structure extension [11, 12], instance selection [57], instance weighting [14], feature selection [17] and feature weighting [9, 20, 22]. Among them, feature weighting approaches [9, 20, 22] have attracted a lot of attention recently, which assign different weights to features to decouple the correlation between features [9, 20, 22]. In [20], attributes and instance are weighted simultaneously. Recently, Wang *et al.* developed a regularized attribute weighting framework to automatically balance the generalization ability and discrimination power of NB classifier [9]. These methods partially alleviate the problem, but still not well model the feature correlation.

Artificial defects commonly exist in real-world applications, e.g., missing values or noisy samples. To handle noisy samples and extract the intrinsic data characteristics, many subspace approaches have been developed to remove the unreliable features and extract the discriminant features [26, 133, 134, 154, 162]. For example, Principal Component Analysis (PCA) is often used for dimensionality reduction by projecting the high-dimensional features into a lower-dimensional space [26, 160, 162, 211]. In literature, auto-encoders have been widely used for filling missing values [212] and denoising [210, 213].

In this paper, we aim to address the following three challenges of naive Bayes: 1) Removing the noisy and unreliable feature dimensions; 2) Modeling the correlation between features so that the subsequent naive Bayes could make better use of the discriminant information residing in features; 3) Boosting the discriminant power of features. To tackle these three challenges, we resort to stacked auto-encoder [214]. Stacked auto-encoder is often trained in a self-supervised manner. A portion of the feature entities are intentionally masked off, and the encoder maps the original feature to a lower-dimensional code to remove the noise and uncover the underlying intrinsic data characteristics. The code is then used to reconstruct the original feature [210, 213], with the target of minimizing the reconstruction error. In such a way, the stack auto-encoder could effectively remove the noise, and embed the discriminant information into the compact codes [210, 213, 215]. Apparently, the correlation between features is embedded into the codes as well, which is beneficial to the subsequent naive Bayes classifier.

To the best of our knowledge, the stacked auto-encoder has never been used for boosting the discriminant power of features. It is often advantageous to map the feature into a higher-dimensional space so that the features can be linearly separable [175]. The stacked auto-encoder, however, often maps the feature into a compact representation, which many result in discriminant information loss. To tackle this problem, we propose a stacked auto-encoder consisting of two encoders: shrink encoder and expansion encoder. The shrink encoder derives a compact feature representation while the expansion encoder maps the derived compact codes into a higher-dimensional space to enhance the discriminant power of the features. Furthermore, by concatenating the learned representation with the original feature and reconstructed one, the classification performance of the subsequent regularized naive Bayes is significantly improved.

The proposed Feature-Augmented Regularized Naive Bayes (FAR-NB) is compared with the state-of-the-art NB classifiers on a set of machine-learning datasets for various applications. It significantly and consistently outperforms all the compared methods. The average performance gain on 20 datasets is 5.71% compared with the second best method, RNB [9].

Our main contributions can be summarized as follows: 1) We propose a feature augmentation method for naive Bayes to exploit the feature correlation and reduce data noise using the stacked auto-encoder. 2) The designed stacked auto-encoder can greatly boost the discriminant power of features by mapping them into a higher-dimensional space, which greatly improves the classification performance. 3) The proposed method is integrated with the regularized naive Bayes and achieves superior performance against state-of-the-art NB classifiers.

6.2 Related Work

Feature extraction methods have been widely utilized to discover the compact feature representations from the raw data, which can be broadly categorized into statistical methods [26, 153, 154] and neural networks [210, 214, 216, 217]. The former include Principal Component Analysis [153, 154], Linear Discriminant Analysis [26] and many others, and the latter include Auto-encoder (AE) [210, 214], Artificial Neural Network [217], Convolutional Neural Network (CNN) [216], and many others.

The auto-encoder encodes the input features in a self-unsupervised way, aiming to derive a compact feature representation by mapping the feature into a lower-dimensional space [215]. There are many variations of AEs, e.g., sparse auto-encoder [218], denoising auto-encoder [213], contractive auto-encoder [219] and convolutional auto-encoder [220]. In literature, the feature learning approaches for naive Bayes are less explored. In [210], an unsupervised feature learning approach is developed for malware classification using the auto-encoder and the performance of naive Bayes classifier has been greatly improved. Recently, Khamparia *et al.* utilized deep stacked auto-encoder for chronic kidney disease classification to learn representative features [214].

6.3 Proposed Feature-Augmented Regularized Naive Bayes

6.3.1 Preliminaries of Regularized Naive Bayes

In the Bayesian classification framework, the posterior probability is defined as:

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c)P(c)}{P(\boldsymbol{x})}, \tag{6.1}$$

where $\boldsymbol{x}$ is the feature vector, c is the classification variable, $P(c)$ is the prior probability, $P(\boldsymbol{x})$ is the evidence, $P(\boldsymbol{x}|c)$ is the likelihood probability distribution and $P(c|\boldsymbol{x})$ the posterior probability. Because it is difficult to reliably estimate the likelihood probability $P(\boldsymbol{x}|c)$ due to the curse of dimensionality, in naive Bayes methods, the likelihood is often estimated by assuming the feature independence,

$$P(\boldsymbol{x}|c) = \prod_{j=1}^{m} P(x_j|c), \tag{6.2}$$

where x_j is the j-th feature dimension of $\boldsymbol{x}$ and m is the feature dimensionality. Despite its simplicity, naive Bayes has shown good performance in many applications [17, 206–210].

Apparently the feature correlation is not modeled in naive Bayes. To address this problem, many feature weighting approaches [9, 20, 22] have been developed. In WANBIA [21], each feature is assigned a different weight to highlight the feature with a large discriminant power,

$$P_I(\boldsymbol{x}|c) = \prod_{j=1}^{m} P(x_j|c)^{\boldsymbol{w}_j}, \tag{6.3}$$

where $\boldsymbol{w}_j$ is the weight for the j-th feature dimension. The weights are optimized by minimizing the mean squared error between the estimated posteriors and the posteriors derived using ground-truth labels. Jiang *et al.* showed that a class-specific weight could further enhance the discrimination power of naive Bayes [22],

$$P_D(\boldsymbol{x}|c) = \prod_{j=1}^{m} P(x_j|c)^{\mathbf{W}_{c,j}}, \tag{6.4}$$

where $\boldsymbol{W}_{c,j}$ is the entry for the weight matrix $\boldsymbol{W}$ for the j-th attribute of the class c. As a result, different weights are assigned to attributes for different classes. Class-specific attribute weights provide more discriminant power, but the model complexity is considerably increased, so the generalization capability may decrease. To tackle this problem, regularized naive Bayes [9] determines the likelihood probability as,

$$P_R(\boldsymbol{x}|c) = \prod_{j=1}^{m} \left((1-\alpha)P_D(x_j|c)^{\boldsymbol{W}_{c,j}} + \alpha P_I(x_j|c)^{\boldsymbol{w}_j}\right), \tag{6.5}$$

where $P_D(x_j|c)$ is the likelihood weighted using the class-dependent weight matrix $\boldsymbol{W}$, $P_I(x_j|c)$ is the likelihood weighted using the class-independent weight vector $\boldsymbol{w}$ and α is the hyper-parameter for balancing these two models. The model parameters $\boldsymbol{M} = \{\boldsymbol{W}, \boldsymbol{w}, \alpha\}$ are optimized using a gradient descent procedure [9]. These weighted naive Bayes [9, 20, 22] utilize attribute weights to emphasize the discriminative features. However, they could not fully exploit the discriminant information between features.

6.3.2 Overall Architecture of the Proposed Method

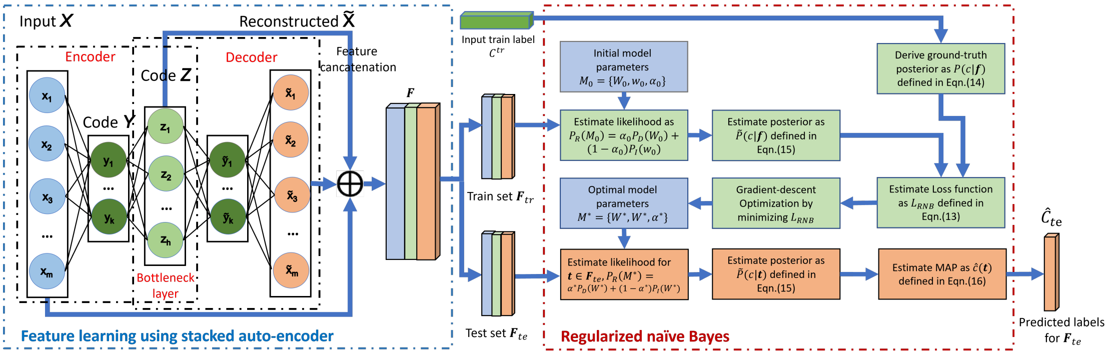

Input X
Reconstructed $\tilde{X}$
Encoder
Code Z
Decoder
Feature cancatenation
Code Y
x_1
x_2
x_3
...
x_m
y_1
y_k
z_1
z_2
z_h
$\tilde{y}_1$
$\tilde{y}_k$
$\tilde{x}_1$
$\tilde{x}_2$
$\tilde{x}_3$
$\tilde{x}_m$
Bottleneck layer
F
Feature learning using stacked auto-encoder
Input train label C^{tr}
Train set F_{tr}
Test set F_{te}
Initial model parameters $M_0 = \{W_0, w_0, \alpha_0\}$
Estimate likelihood as $P_R(M_0) = \alpha_0 P_D(W_0) + (1-\alpha_0)P_I(w_0)$
Estimate posterior as $\tilde{P}(c|\boldsymbol{f})$ defined in Eqn.(15)
Derive ground-truth posterior as $P(c|\boldsymbol{f})$ defined in Eqn.(14)
Optimal model parameters $M^* = \{W^*, W^*, \alpha^*\}$
Gradient-descent Optimization by minimizing L_{RNB}
Estimate Loss function as L_{RNB} defined in Eqn.(13)
Estimate likelihood for $\boldsymbol{t} \in \boldsymbol{F}_{te}, P_R(M^*) = \alpha^* P_D(W^*) + (1-\alpha^*)P_I(W^*)$
Estimate posterior as $\tilde{P}(c|\boldsymbol{t})$ defined in Eqn.(15)
Estimate MAP as $\hat{c}(\boldsymbol{t})$ defined in Eqn.(16)
$\hat{C}_{te}$
Predicted labels for F_{te}
Regularized naïve Bayes

FIGURE 6.1: The overall architecture of the proposed FAR-NB method by resorting an stacked auto-encoder to generate the augmented feature set and then enhance the subsequent regularized naive Bayes for classification.

The proposed method aims to address the following three challenges of previous naive Bayes methods: 1) Noise removal; 2) Encoding the feature correlation; 3) Boosting the discriminant power of naive Bayes. Towards these objectives, we propose a Feature-Augmented Regularized Naive Bayes to learn a discriminant feature representation using an stacked auto-encoder. The overall architecture of the proposed method is shown in Fig. 6.1. It consists of two main stages: unsupervised feature learning using the stacked auto-encoder and the subsequent regularized naive Bayes. The proposed stacked auto-encoder consists of a shrink encoder to derive the compact feature representation and an expansion encoder to boost the discriminant power of the features.

Denote the input features as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i \in \mathbb{R}^m$ is the feature vector for the i-th sample, m is the feature dimensionality and n is the number of instances. To remove the noise and encode the correlation information between features, the shrink encoder is designed to learn a compact feature representation $\boldsymbol{Y} \in \mathbb{R}^{k \times n}$ using all the initial feature dimensions of $\boldsymbol{X}$. Then, the expansion encoder is designed to map $\boldsymbol{Y}$ into higher-dimensional features $\boldsymbol{Z} \in \mathbb{R}^{h \times n}$ to boost the discriminant power. Then, the reconstructed features $\tilde{\boldsymbol{X}}$ are derived from the codes $\boldsymbol{Z}$. The learned features $\boldsymbol{Z}$ are concatenated with the original features $\boldsymbol{X}$ and the reconstructed ones $\tilde{\boldsymbol{X}}$ as the final features.

The stacked auto-encoder is trained in a self-supervised way, in which some feature dimensions of $\boldsymbol{x}$ are intentionally masked off, and the target is to minimize the reconstruction error, towards the objective of removing the noise in data and unveiling the underlying data characteristics. But different from previous stacked auto-encoders [210, 214] that often derive a compact code from the input feature, in our framework, the stacked auto-encoder is designed to boost the discriminant power of features as well by using the expansion encoder. The number of neurons of the inner layers (feature dimensionality k of $\boldsymbol{Y}$ and feature dimensionality m of $\boldsymbol{Z}$) of the stacked auto-encoder is automatically adjusted according to optimally remove the data noise and boost the discriminant power.

Finally, the regularized naive Bayes [9] is trained using the concatenated features as the input. Some preliminaries of the regularized naive Bayes [9] are given in Section 6.3.1. The optimization of the RNB [9] can be found in Section 6.3.5.

6.3.3 Feature Learning Using Stacked Auto-encoder

The designed stacked auto-encoder aims to achieve the following three targets for the subsequent naive Bayes classifier: noise removal, extracting feature correlation and boosting the discriminant power of the model. More specifically, the stacked autoencoder is designed as a feed-forward network to reconstruct $\boldsymbol{X}$ into $\tilde{\boldsymbol{X}}$ with the minimum reconstruction errors. The proposed network contains two encoders: shrink encoder and expansion encoder.

The shrink encoder extracts the intrinsic data characteristics and encodes them into a compact representation, i.e., it maps the input $\boldsymbol{X}$ to $\boldsymbol{Y} \in \mathbb{R}^{k \times n}$, where $k \leq m$ is the number of neuron in the first inner layers,

$$\boldsymbol{Y} = S(\boldsymbol{W}^s \boldsymbol{X} + \boldsymbol{b}^s), \tag{6.6}$$

where $S : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{k \times n}$ is the activation function of the shrink encoder, $\boldsymbol{W}^s \in \mathbb{R}^{k \times m}$ is the weight matrix and $\boldsymbol{b}^s \in \mathbb{R}^k$ is the bias. The activation function is defined as,

$$S(x) = \begin{cases} 0, & if\ x \leq 0, \\ x & if\ 0 < x \leq 1, \\ 1 & if\ x \geq 1. \end{cases} \tag{6.7}$$

The expansion encoder maps the compact feature $\boldsymbol{Y}$ into a higher dimensional space,

$$\boldsymbol{Z} = E(\boldsymbol{W}^e \boldsymbol{Y} + \boldsymbol{b}^e), \tag{6.8}$$

where $E : \mathbb{R}^{k \times n} \rightarrow \mathbb{R}^{h \times n}$ is the activation function of the expansion encoder defined similarly as in Eqn. (6.7), and $h \geq k$. $\boldsymbol{W}^e \in \mathbb{R}^{h \times k}$ is the weight matrix and $\boldsymbol{b}^e \in \mathbb{R}^h$ is the bias.

During the decoding phase, the encoded feature representations $\boldsymbol{Z}$ are transformed back into the original feature space to derive the reconstructed features $\tilde{\boldsymbol{Y}}$,

$$\tilde{\boldsymbol{Y}} = D(\boldsymbol{W}^d \boldsymbol{Z} + \boldsymbol{b}^d), \tag{6.9}$$

where the logistic sigmoid function is used for decoding,

$$D(z) = \frac{1}{1 + e^{-z}}. \tag{6.10}$$

Then $\tilde{\boldsymbol{Y}}$ is similarly transformed back to $\tilde{\boldsymbol{X}}$.

The auto-encoder is trained to minimize the Mean Square Error (MSE) between the input $\boldsymbol{X}$ and the reconstructed $\tilde{\boldsymbol{X}}$,

$$L_{AE} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \tilde{x}_{ij})^2. \tag{6.11}$$

In the traditional stacked auto-encoder, all the encoders are shrink encoders, aiming to derive a compact feature representation so that the unreliable classification information could be removed and the discriminant information embedded across features can be encoded into $\boldsymbol{Z}$. However, some discriminant information may be lost during this process.

6.3.4 Boosting Discriminant Power of Regularized Naive Bayes

To boost the discriminant power of the regularized naive Bayes, we propose to map the compact codes into a higher-dimensional space using the expansion encoder. It remains an open question to determine the optimal feature dimensionalities k of $\boldsymbol{Y}$ and h of $\boldsymbol{Z}$, as they are affected by many factors. 1) The number of training samples n. When n is small, there are insufficient samples to train a reliable network, and hence a smaller network is preferred, i.e., k and h should be kept small. 2) The number of classes. Intuitively, when the number of classes is large, more training samples are needed to reliably estimate the data distribution of each class. Given a fixed number of training samples, we hence prefer a simpler network, i.e., k and h should be smaller. 3) If the input feature dimensionality m is large, there is probably a large amount of redundant information residing in features, and hence we prefer to compress the features into a smaller k-dimensional space, and a slightly larger h to boost the discriminant power. 4) If m is relatively small, we prefer to maintain k similar but smaller than m and then map the compact codes into a slightly higher h-dimensional space so that the features of different classes are linearly separable. The optimal pair of (k, h) is determined empirically in experiments.

The learned feature representation $\boldsymbol{Z}$, the original features $\boldsymbol{X}$ and the reconstructed $\tilde{\boldsymbol{X}}$ all contain discriminant information in different feature spaces. To make full use of all the available discriminant information, we propose to fuse them by concatenating them into the final feature representation as,

$$\boldsymbol{F} = \boldsymbol{X} \oplus \boldsymbol{Z} \oplus \tilde{\boldsymbol{X}}. \tag{6.12}$$

6.3.5 Optimizing Regularized Naive Bayes

The concatenated features $\boldsymbol{F}$ are split into the training set $\boldsymbol{F}_{tr}$ and the testing set $\boldsymbol{F}_{te}$. During the training process, the following loss function is used to optimize the regularized naive Bayes,

$$L_{RNB} = \frac{1}{2} \sum_{\boldsymbol{f}_i \in \boldsymbol{F}_{tr}} \sum_{c} (P(c|\boldsymbol{f}_i) - \tilde{P}(c|\boldsymbol{f}_i))^2, \tag{6.13}$$

where $P(c|\boldsymbol{f}_i)$ is the posterior derived from the ground-truth labels,

$$P(c|\boldsymbol{f}_i) = \begin{cases} 1 & if\ c = c_j, \\ 0 & otherwise. \end{cases} \tag{6.14}$$

$\tilde{P}(c|\boldsymbol{f}_i)$ is the estimated posterior with the regularized likelihood function defined in Eqn. (6.5),

$$\tilde{P}(c|\boldsymbol{f}_i) = P(c)P_R(\boldsymbol{f}_i|c)/P(\boldsymbol{f}_i). \tag{6.15}$$

The optimal model parameters $\boldsymbol{M}^* = \{\alpha^*, \boldsymbol{W}^*, \boldsymbol{w}^*\}$ of the regularized naive Bayes are derived by minimizing the loss function defined in Eqn. ((6.13)) using a gradient-descent-based optimization procedure. More details can be found in [9].

During testing, the posterior probability $\hat{P}(c|\boldsymbol{t})$ for a given testing instance $\boldsymbol{t} \in \boldsymbol{F}_{te}$ is estimated by using Eqn. (6.15) with the optimal model $\boldsymbol{M}^*$. Finally, the class label for each $\boldsymbol{t} \in \boldsymbol{F}_{te}$ is derived by using the MAP estimation as follows:

$$\hat{c}(\boldsymbol{t}) = \arg\max_{c \in \boldsymbol{C}} \hat{P}(c|\boldsymbol{t}), \tag{6.16}$$

where $\boldsymbol{C}$ is the set of labels for all classes.

6.4 Experimental Results

6.4.1 Experimental Settings

The proposed FAR-NB is compared with state-of-the-art NB classifiers including RNB [9], WANBIA [21], CAWNB [22] and AIWNB [20], as summarized in Table 6.1. The experiments are conducted on a collection of benchmark datasets from the University of California at Irvine (UCI) repository [2], which contains a wide range of domains such as medical, business and biology. The number of instances is distributed between 150 and 10992 and the number of attributes varies between 2 and 60. These 20 machine-learning datasets can provide a comprehensive evaluation of the effectiveness of the proposed method. More details of these datasets are described in Tables 6.2. The classification accuracy of each algorithm is derived using 10-fold cross-validation.

Table 6.1: Summary of compared naive Bayes classifiers.

Algorithm	**Description**
RNB [9]	Wrapper-based regularized attribute weighting method
CAWNB [22]	Wrapper-based class-specific attribute weighting method
WANBIA [21]	Wrapper-based class-independent attribute weighting method
AIWNB [20]	Filter-based attribute and instance weighting method, either eager learning AIWNB^E or lazy learning AIWNB^L

6.4.2 Ablation Study

For an ablation study, the proposed method is compared with the following methods:

Original Features: The original feature is fed into the regularized naive Bayes [9] for classification. This comparison could demonstrate the effectiveness of the proposed feature augmentation method in contrast to using the original features.

[2] https://archive.ics.uci.edu/ml/index.php

Table 6.2: The datasets are collected from real-world applications in various domains. The number of instances varies between 150 and 10992 and the feature dimensionalities are distributed between 2 and 60.

	Inst.	Attr.	Class	Domain
Balance	625	4	3	Social
Banana	5300	2	2	Artificial
Banknote	1372	5	2	Business
Bupa	345	6	2	Medical
Clevland	303	13	5	Medical
Contraceptive	1473	9	3	Medical
Ecoli	336	7	8	Biology
Hayes	160	4	3	Social
Iris	150	4	3	Biology
Mammographic	961	5	2	Medical
Newthyroid	215	5	3	Medical
Penbased	10992	16	10	Artificial
Satimage	6435	36	7	Medical
Segment	2310	19	7	Artificial
Sonar	208	60	2	Physical
Specfheart	267	44	2	Physical
Tae	151	5	3	Education
Vowel	990	13	11	Artificial
Wine	178	13	3	Chemical
Yeast	1484	8	10	Biology

Baseline: The stacked auto-encoder [215] is chosen as the baseline method to derive a compact feature representation and the derived features are fed into the regularized naive Bayes [9] for classification. The feature dimension of the bottleneck layer is empirically set to half of the input feature dimensionality. The comparison to this baseline can show the power of the proposed feature augmentation method, in contrast to compressing the input feature as in most existing auto-encoders [210, 214, 215].

As shown in Table 6.3, FAR-NB achieves the highest classification performance on all datasets in comparison to using the original features and the compact feature representation derived using the traditional stacked auto-encoder [215]. Compared with the original features, the average classification accuracy for the compact features has been greatly reduced by more than 7%. It shows that directly applying the traditional stacked auto-encoder could not produce good performance. The proposed method utilizes the stacked auto-encoder in a very different way, which greatly boost the discriminant power of the model and hence significantly improves the classification accuracy by 13.27% on

Table 6.3: Classification accuracy of the proposed FAR-NB comparing with RNB and baseline method in which the auto-encoder is used to derive a compact feature representation for regularized naive Bayes.

	Original Features	Baseline	FAR-NB
Balance	0.7186	0.6703	**0.8815**
Banana	0.7338	0.4483	**0.8621**
Banknote	0.9278	0.8053	**0.9854**
Bupa	0.5327	0.5798	**0.5882**
Clevland	0.5773	0.5619	**0.6237**
Contraceptive	0.5234	0.4243	**0.5485**
Ecoli	0.8339	0.6640	**0.8430**
Hayes	0.6003	0.6101	**0.7750**
Iris	0.9333	0.9467	**0.9600**
Mammographic	0.8263	0.6671	**0.8419**
Newthyroid	0.9535	0.8974	**0.9621**
Penbased	0.9311	0.9097	**0.9542**
Satimage	0.8577	0.8684	**0.8699**
Segment	0.9459	0.8333	**0.9593**
Sonar	0.7742	0.6727	**0.7983**
Specfheart	0.8114	0.7820	**0.8269**
Tae	0.3440	0.3440	**0.4683**
Vowel	0.6465	0.5616	**0.8192**
Wine	0.9719	0.8595	**0.9941**
Yeast	0.5729	0.3982	**0.5965**
AVG	0.7508	0.6752	**0.8079**

average. These demonstrate the effectiveness of the proposed feature augmentation approach over the traditional stacked auto-encoder.

6.4.3 Comparisons to State-of-the-art Naive Bayes Classifiers

The comparisons to the state-of-the-art NB methods on 20 benchmark datasets are summarized in Table 6.4. The average classification accuracy of each algorithm over the datasets is summarized at the bottom of Table 6.4, which provides a straightforward comparison of different approaches. To measure the significance of the performance gain, a paired one-tailed t-test with $p = 0.05$ significance level is deployed. $W/T/L$ values over all datasets are presented at the bottom of Table 6.4, indicating that the proposed method wins on W datasets, ties on T datasets and loses on L datasets.

As shown in Table 6.4, the proposed FAR-NB consistently outperforms all the compared methods on all the datasets. Among them, FAR-NB is significantly better than

Table 6.4: Comparisons between FAR-NB and other state-of-the-art NB methods. The proposed FAR-NB significantly and consistently outperforms all the compared methods on all the datasets. On average, the performance gain of FAR-NB is 5.71% compared with the previous best method, RNB [9].

	FAR-NB	RNB [9]	CAWNB [22]	WANBIA [21]	AIWNB^E [20]	AIWNB^L [20]
Balance	**0.8815**	0.7186	0.7186	0.7186	0.7153	0.7008
Banana	**0.8621**	0.7338	0.7338	0.7283	0.7198	0.7332
Banknote	**0.9854**	0.9278	0.9278	0.9213	0.9206	0.9257
Bupa	**0.5882**	0.5327	0.5327	0.5327	0.4202	0.4202
Clevland	**0.6237**	0.5773	0.5845	0.5773	0.5717	0.5815
Contraceptive	**0.5485**	0.5234	0.5179	0.5139	0.5072	0.5112
Ecoli	**0.8430**	0.8339	0.8338	0.8251	0.8223	0.8223
Hayes	**0.7750**	0.6003	0.6003	0.6003	0.6003	0.6003
Iris	**0.9600**	0.9333	0.9333	0.9333	0.9267	0.9267
Mammographic	**0.8419**	0.8263	0.8252	0.8252	0.8242	0.8232
Newthyroid	**0.9621**	0.9535	0.9535	0.9580	0.9576	0.9532
Penbased	**0.9542**	0.9311	0.9289	0.8988	0.8882	0.9360
Satimage	**0.8699**	0.8577	0.8420	0.8440	0.8140	0.8544
Segment	**0.9593**	0.9459	0.9381	0.9472	0.9264	0.9420
Sonar	**0.7983**	0.7742	0.7699	0.7837	0.7649	0.7697
Specfheart	**0.8269**	0.8114	0.7856	0.7854	0.7507	0.7507
Tae	**0.4683**	0.3440	0.3440	0.3440	0.3244	0.3244
Vowel	**0.8192**	0.6465	0.6364	0.6414	0.6364	0.6687
Wine	**0.9941**	0.9719	0.9719	0.9830	0.9771	0.9660
Yeast	**0.5965**	0.5729	0.5756	0.5675	0.5715	0.5715
AVG	0.8079	0.7508	0.7477	0.7464	0.7320	0.7391
W/T/L	-	12/8/0	13/7/0	13/7/0	15/5/0	15/5/0

RNB, CAWNB, WANBIA, AIWNB^E and AIWNB^L on 12, 13, 13, 15 and 15 datasets, respectively. Compared with wrapper-based attribute weighting methods, e.g. RNB, CAWNB and WANBIA, the proposed FAR-NB obtains the performance gain of 5.71%, 6.02% and 6.15% on average, respectively. Compared with filter-based AIWNB^E and AIWNB^L, FAR-NB achieves improvements of 7.59% and 6.88% for the average classification accuracy over 20 datasets. These demonstrate the effectiveness of the proposed feature augmentation method.

For a better visualization, the performance gain of FAR-NB over the second best performed method, RNB [9], on each dataset is shown in Fig. 6.2. FAR-NB obtains more than 10% of improvement for classification accuracy on 5 datasets, e.g., ‘Banana’, ‘Balance’, ‘Hayes’, ‘Tae’ and ‘Vowel’. Besides, FAR-NB can achieve more than 2% of performance gain compared with RNB [9] on most datasets.

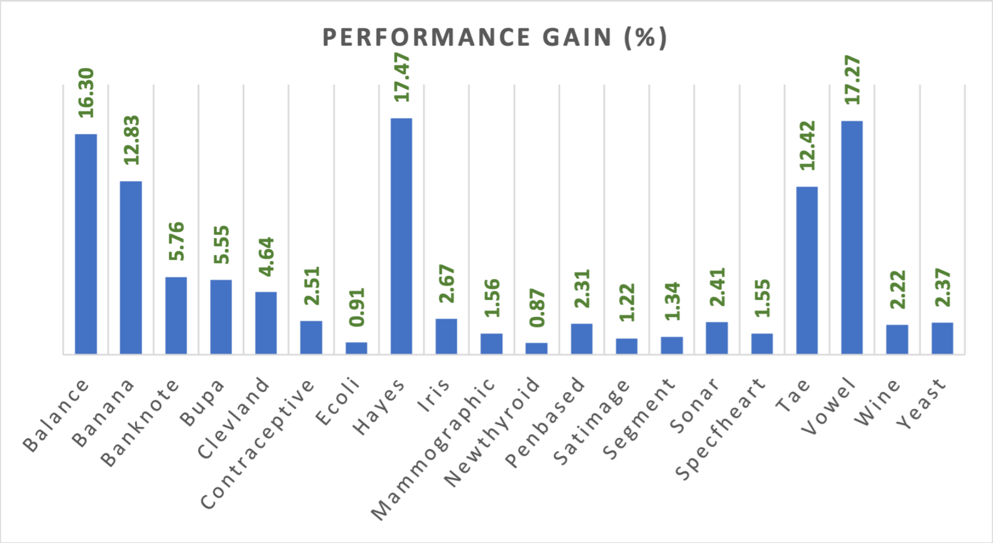


FIGURE 6.2: The performance gain of the proposed FAR-NB on each dataset compared to RNB [9].

6.5 Summary

The performance of naive Bayes is often limited by lack of the correlation information between features. Many approaches have been developed to alleviate this problem, e.g., feature weighting methods. But these approaches could not fully exploit the discriminant information between features. In this paper, we propose a feature augmentation method for the regularized naive Bayes to extract the discriminant information between features, reduce data noise and boost the discriminant power of the model. Towards these objectives, we resort to the stacked auto-encoder. Different from traditional stacked auto-encoders that map the original features into compact codes, the proposed FAR-NB consists of two encoders, one removes the noise and unreliable information, and another maps the derived compact code into a higher-dimensional space to boost the discriminant power of the model. To further boost the classification performance, the derived features are concatenated with the original features and the reconstructed ones as the augmented features. The proposed feature augmentation method is integrated with the regularized naive Bayes. It is compared with state-of-the-art NB classifiers on 20 datasets for various applications. Experimental results demonstrate that the proposed FAR-NB consistently and significantly outperforms all the compared NB classifiers on all datasets.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we study the naive Bayes classification framework and aim to address the independent assumption from different perspectives. This chapter is organized as follows. In Section 7.1.1, our contributions of naive Bayes classifier on regularized attribute weighting framework are presented. The contributions of achieving a better trade-off between generalization ability and discrimination power on discretization for naive Bayes classifier is given in Section 7.1.2. The contributions of exploiting the discriminant information in the data by feature augmentation framework for naive Bayes classifiers are discussed in Section 7.1.3. The comparative study between the proposed methods is described in Section 7.1.4. Finally, Some potential research directions are discussed in Section 7.2.

7.1.1 Contributions on regularizing attribute-weighting framework on naive Bayes

Recently in literatures, we find that class-dependent attribute-weighting naive Bayes has poor generalization capabilities on relatively small datasets. Therefore, we propose to add a regularization term to alleviate the problem. The regularization term is extracted from a simpler naive Bayes which has better generalization capabilities. The proposed regularized naive Bayes (RNB) is hence derived by integrating the regularization term into the class-specific attribute weighted naive Bayes method. A gradient-descent-based optimization procedure has been designed to derive the optimal model parameters including class-dependent weight matrix $\boldsymbol{W}$, class-independent weight vector $\boldsymbol{w}$ and the hyper-parameter α. We test various naive Bayes classifiers and RNB demonstrates a superior performance to others.

7.1.2 Contributions on discretization methods for naive Bayes

Naive Bayes methods often utilize the discretization method to improve the efficiency and generalization ability of the classification model. Most discretization methods only exploit the data characteristics based on the labeled data while ignoring the amount of unlabeled data. To better utilize the overall discriminant information, a semi-supervised discretization framework is designed to boost the discrimination power of naive Bayes classifiers. A pseudo-labeling technique is first utilized to derive the pseudo labels for unlabeled data. Then, an adaptive discriminative discretization is introduced to strategically lower the threshold of selection criterion in MDLP and reduce the information loss during the discretization process. Finally, the proposed semi-supervised discretization method is integrated with state-of-the-art naive Bayes classifiers and greatly enhanced their performance.

Another problem of previous data discretization methods is that they often overemphasize maximizing the discriminant information while overlooking the primary goal of data discretization in classification, *i.e.*, to enhance the generalization ability of the classifier. To address this problem, a Maximal-Dependency-Minimal-Divergence scheme is proposed to simultaneously maximize the generalization capability and discriminant information. The proposed MDmD criterion is difficult to implement in practice due to the difficulty in estimating the high-order mutual information. We hence proposed a more practical solution, Maximal-Relevance-Minimal-Divergence criterion, which discretizes one attribute at a time in a top-down manner. The proposed MRmD criterion generates a discretization scheme with a trade-off between retaining the discriminant information and improving the generalization ability for the subsequent classifier.

7.1.3 Contributions on feature augmentation method for naive Bayes

As known, the performance of naive Bayes is often limited by lack of the correlation information between features. Many approaches have been developed to alleviate this problem, e.g., feature weighting methods. But these approaches could not fully exploit the discriminant information between features. In this paper, we propose a feature augmentation method for the naive Bayes to extract the discriminant information between features, reduce data noise and boost the discriminant power of the model. Toward these objectives, we resort to the stacked auto-encoder. Different from traditional stacked auto-encoders that map the original features into compact codes, the proposed FAR-NB consists of two encoders, one removes the noise and unreliable information, and another maps the derived compact code into a higher-dimensional space to boost the discriminant power of the model. To further boost the classification performance, the derived features are concatenated with the original features and the reconstructed ones as the augmented features. The proposed feature augmentation method is integrated with the regularized naive Bayes.

7.1.4 Comparison of the proposed improvements on naive Bayes

To analyze the superior of the proposed methods, the comparison between the original naïve Bayes and the proposed methods has been shown in Table 7.1 including Regularized Naive Bayes (RNB) [9], Semi-supervised Adaptive Discriminative Discretization for Naive Bayes (SADD-NB) [177], Maximal-Relevancy-Minimal-Divergence for Naive Bayes (MRmD-NB) [196], and Feature Augmented Naive Bayes (FA-NB) [184]. In the experiment, 45 machine-learning benchmark datasets are used for comprehensive evaluation and the description of these datasets is shown in Table 6.2. The classification accuracy of each algorithm is derived using 10-fold cross-validation. As shown in Table 7.1, the proposed RNB, SADD-NB, MRmD-NB and FA-NB outperform the naive Bayes with the performance gain of 2.84%, 3.92%, 4.22% and 1.74% on average classification accuracy, respectively. Among them, the proposed MRmD achieves the highest classification accuracy indicating its robust ability to handle diverse data by focusing on maximizing discriminant information and generalization ability during discretization.

Table 7.1: Comparisons of the proposed methods including Regularized attribute-weighting NB, NB based on Semi-supervised Adaptive Discriminative Discretization (SADD-NB), NB based on Max-Relevancy-Min-Divergence discretization (MRmD-NB) and Feature Augmented naïve Bayes (FA-NB).

	NB	RNB	SADD-NB	MRmD-NB	FA-NB
abalone	0.2496	**0.2674**	0.2537	0.2554	0.2528
appendicitis	0.8709	0.8755	0.8682	**0.8791**	0.8591
australian	0.8449	**0.8680**	0.8535	0.8637	0.8492
auto	0.6732	**0.8244**	0.7402	0.7693	0.7105
balance	0.7266	0.7186	0.8784	**0.9104**	0.7568
banana	0.7247	0.7338	0.7198	0.7296	**0.7519**
bands	0.5045	0.7069	**0.7605**	0.7364	0.7087
banknote	0.9205	0.9278	**0.9322**	0.9155	0.9118
bupa	0.5715	0.5327	0.6576	**0.6842**	0.5824
clevland	0.5545	**0.5857**	0.5611	0.5807	0.5769
climate	0.9352	0.9426	0.9259	0.9351	**0.9426**
contraceptive	0.5051	**0.5227**	0.5194	0.5221	0.5180
crx	0.8565	**0.8652**	0.8507	0.8623	0.8551
dermatology	0.9782	**0.9864**	0.9807	0.9863	0.9784
ecoli	0.8216	0.8339	**0.8662**	0.8428	0.8400
flare-solar	0.6754	0.6820	**0.6885**	0.6829	0.6800
glass	0.7206	0.7197	0.7429	**0.7567**	0.7246
haberman	0.7285	0.7318	0.7451	**0.7480**	0.7318
hayes	0.5202	0.6003	**0.8220**	0.8009	0.6736
heart	0.8407	**0.8519**	0.8370	0.8407	0.8370
hepatitis	0.8383	0.8404	0.8533	**0.8654**	0.8463
iris	0.9267	0.9333	**0.9600**	0.9400	0.9400
mammographic	0.8221	0.8263	0.8305	**0.8336**	0.8232
movement	0.6056	0.6875	**0.7777**	0.6890	0.7104
newthyroid	0.9489	0.9580	0.9485	**0.9766**	0.9628
pageblocks	0.9311	**0.9633**	0.9393	0.9410	0.9401
penbased	0.8766	**0.9311**	0.8843	0.8867	0.8764
phoneme	0.7689	**0.8022**	0.7765	0.7913	0.7789
pima	0.7526	0.7486	**0.7682**	0.7461	0.7538
saheart	0.6624	0.7012	0.6903	**0.7079**	0.6710
satimage	0.8210	**0.8577**	0.8245	0.8228	0.8283
segment	0.9104	**0.9459**	0.9372	0.9229	0.9216
seismic	0.8200	0.9342	0.8371	**0.9342**	0.8475
sonar	0.7688	0.7742	**0.8023**	0.7840	0.7699
spambase	0.8989	**0.9394**	0.9018	0.9053	0.9094
specfheart	0.7305	**0.8114**	0.7561	0.7971	0.7403
tae	0.3442	0.3440	0.5024	**0.5657**	0.3442
thoracic	0.8213	**0.8362**	0.8191	0.8298	0.8213
titanic	0.7760	0.7760	0.7787	**0.7819**	0.7506
transfusion	0.7500	0.7621	0.7474	**0.7794**	0.7447
vehicle	0.5910	**0.6774**	0.6372	0.6337	0.6195
vowel	0.6030	0.6465	**0.7576**	0.6404	0.6485
wine	**0.9886**	0.9830	0.9830	0.9830	0.9775
wisconsin	0.9728	0.9722	0.9736	**0.9736**	0.9721
yeast	0.5695	0.5729	**0.5979**	0.5883	0.5647
MEAN	0.7494	0.7778	0.7886	0.7916	0.7668

SADD-NB performs closely to the best, leveraging semi-supervised learning to achieve remarkable accuracy gains, particularly effective in exploiting the discriminant information residing in unlabeled data. RNB shows consistent improvements by incorporating a simple and effective regularization framework, thus mitigating overfitting and enhancing generalization. FA-NB exhibits the promising improvement and validates naive Bayes could benefit from augmented feature representations. Overall, these enhanced methods provide significant advancements over the original NB classifier, with MRmD and SADD emerging as the most effective enhancements, highlighting their potential for broader application in various classification tasks.

7.2 Future work

In the future, we plan to further improve the robustness of naive Bayes classifiers and apply the proposed techniques to real-world applications.

7.2.1 Simultaneous discretization and feature selection framework for naive Bayes

Data discretization and feature selection are two data reduction techniques in the field of machine learning, pattern recognition and data mining. Feature selection methods have been explored over the last decades to reduce the noise and redundancy in feature sets and improved classification performance. However, existing researches rarely consider discretization and feature selection simultaneously. We have already proposed an information-based discretization method. By combining discretization with feature selection, there are two main directions: a) Combine MRmD with existing feature selection methods: Information theory is widely used in feature selection techniques such as Maximal-Relevance-Minimal-Redundancy (mRMR) and Conditional Mutual Information Maximization (CMIA). Thus, we can directly combine MRMG with mRMR or CMIA to make the data more concise and effective. b) Design a discretization-based feature selection method: Data discretization and feature selection interact with each other during the selection process. Therefore, we can select the discretization scheme and feature subset simultaneously by designing an information-based selection criterion.

7.2.2 Wrapper-based discretization for naive Bayes

Traditional feature selection and discretization often utilize the greedy search to find the solution which may result in the local optimum. To approach the global optimal solution, many evolutionary algorithms are applied by designing a set of objective functions. Thus, the proposed MRmD discretization can be turned into an optimization problem by defining the objective function to maximize the discriminant information and generalization ability. Then, the evolutionary methods, *e.g.*, the genetic algorithm and particle swarm optimization, can be applied to jointly consider the whole feature space. Hence, the derived discretization scheme could enhance the performance of naive Bayes classifiers.

Chapter 8

Appendix

8.1 Formulation for RNB

In this section, a brief derivation of the gradients of f w.r.t $\boldsymbol{W}$ and $\boldsymbol{w}$ is provided. Firstly, the partial derivative of f w.r.t. each element of $\boldsymbol{W}$, $w_{c,j}$, is calculated as:

$$\frac{\partial f}{\partial w_{c,j}} = -\alpha \sum_{\boldsymbol{x} \in D} \Big(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x}) \Big) \frac{\partial \hat{P}_D(c|\boldsymbol{x})}{\partial w_{c,j}}. \tag{8.1}$$

Denote $\gamma_D(\boldsymbol{W}) = \pi_c \prod_j \theta_{c,j}^{w_{c,j}}$. Then, $\hat{P}_D(c|\boldsymbol{x})$ defined in (3.7) can be re-written as $\hat{P}_D(c|\boldsymbol{x}) = \frac{\gamma_D(\boldsymbol{W})}{\sum_{c'} \gamma_D(\boldsymbol{W})}$. It is easy to show that

$$\frac{\partial \hat{P}_D(c|\boldsymbol{x})}{\partial \gamma_D(\boldsymbol{W})} = \frac{\sum_{c' \neq c} \gamma_D(\boldsymbol{W})}{(\sum_{c'} \gamma_D(\boldsymbol{W}))^2}, \tag{8.2}$$

$$\frac{\partial \gamma_D(\boldsymbol{W})}{\partial w_{c,j}} = \gamma_D(\boldsymbol{W}) \log(\theta_{c,j}). \tag{8.3}$$

Derive $\frac{\partial \hat{P}_D(c|\boldsymbol{x})}{\partial w_{c,j}}$ using the chain rule by utilizing (8.2) and (8.3), and then plug it into (8.1) to obtain the partial derivative of f w.r.t. $w_{c,j}$ as defined in (3.15).

Secondly, the partial derivative of f w.r.t. w_j is derived as:

$$\frac{\partial f}{\partial w_j} = -(1-\alpha) \sum_{\boldsymbol{x} \in D} \sum_{c} \Big(P(c|\boldsymbol{x}) - \hat{P}(c|\boldsymbol{x}) \Big) \frac{\partial \hat{P}_I(c|\boldsymbol{x})}{\partial w_j}. \tag{8.4}$$

Denote $\gamma_I(\boldsymbol{w}) = \pi_c \prod_j \theta_{c,j}^{w_j}$. Similarly, $\hat{P}_I(c|\boldsymbol{x})$ defined in (3.8) can be re-written as $\hat{P}_I(c|\boldsymbol{x}) = \frac{\gamma_I(\boldsymbol{w})}{\sum_{c'} \gamma_I(\boldsymbol{w})}$. Note that every term in the summation of the denominator is a function of w_j. The partial derivative $\frac{\partial \hat{P}_I(c|\boldsymbol{x})}{\partial w_j}$ is calculated as:

$$\frac{\partial \hat{P}_I(c|\boldsymbol{x})}{\partial w_j} = \frac{1}{\sum_{c'} \gamma_I(\boldsymbol{w})} \left(\frac{\partial \gamma_I(\boldsymbol{w})}{\partial w_j} - \hat{P}_I(c|\boldsymbol{x}) \sum_{c'} \frac{\partial \gamma_I(\boldsymbol{w})}{\partial w_j} \right)$$

Similar to (8.3), it is easy to show that $\frac{\partial \gamma_I(\boldsymbol{w})}{\partial w_j} = \gamma_I(\boldsymbol{w}) \log(\theta_{c,j})$. Plug it into (8.4), the partial derivative of f w.r.t. w_j shown in (3.16) can be obtained.

Author’s Publications

Journal Papers

- **Shihe Wang**, Jianfeng Ren, and Ruibin Bai. “A regularized attribute weighting framework for naive Bayes.” IEEE Access 8 (2020): 225639-225649.
- **Shihe Wang**, Jianfeng Ren, and Ruibin Bai. “A semi-supervised adaptive discriminative discretization method improving discrimination power of regularized naive Bayes.” Expert Systems with Applications (2023): 120094.
- **Shihe Wang**, Jianfeng Ren, Ruibin Bai, Yuan Yao, and Xudong Jiang. “A Max-relevance-min-divergence Criterion for Data Discretization with Applications on Naive Bayes.” Pattern Recognition 149 (2024): 110236.
- **Shihe Wang**, Jianfeng Ren, Ruibin Bai, Yuan Yao, and Xudong Jiang. “Boosting the Discriminant Power of Naive Bayes via Cascaded Feature Augmentation” Submitted to Pattern Recognition (2024).

Conference Papers

- **Shihe Wang**, Jianfeng Ren, Xiaoyu Lian, Ruibin Bai, and Xudong Jiang. “Boosting the Discriminant Power of Naive Bayes.” In 2022 26th International Conference on Pattern Recognition (ICPR), pp. 4906-4912. IEEE, 2022.
- Chenglin Yao, **Shihe Wang**, Jialu Zhang, Wentao He, Heshan Du, Jianfeng Ren, Ruibin Bai, and Jiang Liu. “rPPG-based spoofing detection for face mask attack using efficientnet on weighted spatial-temporal representation.” In 2021 IEEE International Conference on Image Processing (ICIP), pp. 3872-3876. IEEE, 2021.
- Xingke Song, Jiahuan Jin, Chenglin Yao, **Shihe Wang**, Jianfeng Ren, and Ruibin Bai, “Siamese-discriminant deep reinforcement learning for solving jigsaw puzzles with large eroded gaps,” in AAAI Conference on Artificial Intelligence (AAAI), 2023.

Bibliography

[1] Narges Sharif-Razavian and Andreas Zollmann. An overview of nonparametric Bayesian models and applications to natural language processing. *Science*, pages 71–93, 2008.

[2] Zezhou Cheng, Matheus Gadelha, Subhransu Maji, and Daniel Sheldon. A bayesian perspective on the deep image prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5443–5451, 2019.

[3] Wenjian Xu, Xuanshi Liu, Fei Leng, and Wei Li. Blood-based multi-tissue gene expression inference with bayesian ridge regression. *Bioinformatics*, 36(12):3788–3794, 2020.

[4] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A Bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, 18(70):1–37, 2017.

[5] S. Ruan, H. Li, C. Li, and K. Song. Class-specific deep feature weighting for naïve Bayes text classifiers. *IEEE Access*, 8:20151–20159, 2020.

[6] Evanson Mwangi Karanja, Shedden Masupe, and Mandu Gasennelwe Jeffrey. Analysis of internet of things malware using image texture features and machine learning techniques. *Internet of Things*, 9:100153, 2020.

[7] Chunying Zhang, Xueming Duan, Fengchun Liu, Xiaoqi Li, and Shouyue Liu. Three-way naive bayesian collaborative filtering recommendation model for smart city. *Sustainable Cities and Society*, 76:103373, 2022.

[8] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943, 2021.

[9] S. Wang, J. Ren, and R. Bai. A regularized attribute weighting framework for naive Bayes. *IEEE Access*, 8:225639–225649, 2020.

[10] Jie Wang, Jianqing Liang, Junbiao Cui, and Jiye Liang. Semi-supervised learning with mixed-order graph convolutional networks. *Information Sciences*, 573:171–181, 2021.

[11] Jia Wu, Shirui Pan, Xingquan Zhu, Peng Zhang, and Chengqi Zhang. Sode: Self-adaptive one-dependence estimators for classification. *Pattern Recognition*, 51:358–377, 2016.

[12] Liangxiao Jiang, Shasha Wang, Chaoqun Li, and Lungan Zhang. Structure extended multinomial naive Bayes. *Information Sciences*, 329:346–356, 2016.

[13] Shasha Wang, Liangxiao Jiang, and Chaoqun Li. Adapting naive Bayes tree for text classification. *Knowledge and Information Systems*, 44(1):77–89, 2015.

[14] Wenqiang Xu, Liangxiao Jiang, and Liangjun Yu. An attribute value frequency-based instance weighting filter for naive Bayes. *Journal of Experimental & Theoretical Artificial Intelligence*, 31(2):225–236, 2019.

[15] Liangxiao Jiang, Zhihua Cai, Harry Zhang, and Dianhong Wang. Naive Bayes text classifiers: a locally weighted learning approach. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(2):273–286, 2013.

[16] Armin Askari, Alexandre d'Aspremont, and Laurent El Ghaoui. Naive feature selection: Sparsity in naive Bayes. In *International Conference on Artificial Intelligence and Statistics*, pages 1813–1822. PMLR, 2020.

[17] Bo Tang, Steven Kay, and Haibo He. Toward optimal feature selection in naive Bayes for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 28(9): 2508–2521, 2016.

[18] Liangxiao Jiang, Ganggang Kong, and Chaoqun Li. Wrapper framework for test-cost-sensitive feature selection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[19] Liangxiao Jiang, Lungan Zhang, Chaoqun Li, and Jia Wu. A correlation-based feature weighting filter for naive Bayes. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):201–213, 2018.

[20] Huan Zhang, Liangxiao Jiang, and Liangjun Yu. Attribute and instance weighted naive Bayes. *Pattern Recognition*, 111:107674, 2021.

[21] Nayyar A Zaidi, Jesús Cerquides, Mark J Carman, and Geoffrey I Webb. Alleviating naive Bayes attribute independence assumption by attribute weighting. *The Journal of Machine Learning Research*, 14(1):1947–1988, 2013.

[22] Liangxiao Jiang, Lungan Zhang, Liangjun Yu, and Dianhong Wang. Class-specific attribute weighted naive Bayes. *Pattern Recognition*, 88:321–330, 2019.

[23] Lukasz A Kurgan and Krzysztof J Cios. CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):145–153, 2004.

[24] Alberto Cano, José María Luna, Eva L Gibaja, and Sebastián Ventura. LAIM discretization for multi-label data. *Information Sciences*, 330:370–384, 2016.

[25] Cheng-Jung Tsai, Chien-I Lee, and Wei-Pang Yang. A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences*, 178(3):714–731, 2008.

[26] Xudong Jiang. Asymmetric principal component and discriminant analyses for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5): 931–937, 2008.

[27] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[28] Nasir Saeed, Haewoon Nam, Mian Imtiaz Ul Haq, and Dost Bhatti Muhammad Saqib. A survey on multidimensional scaling. *ACM Computing Surveys (CSUR)*, 51(3):1–25, 2018.

[29] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954. PMLR, 2017.

[30] Bradley Efron. Bayes' theorem in the 21st century. *Science*, 340(6137):1177–1178, 2013.

[31] Christopher KI Williams and David Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12): 1342–1351, 1998.

[32] Liangxiao Jiang, Chaoqun Li, Shasha Wang, and Lungan Zhang. Deep feature weighting for naive Bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, 52:26–39, 2016.

[33] Mark Hall. A decision tree-based attribute weighting filter for naive Bayes. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 59–70. Springer, 2006.

[34] Chang-Hwan Lee, Fernando Gutierrez, and Dejing Dou. Calculating feature weights in naive Bayes with Kullback-Leibler measure. In *2011 IEEE 11th International Conference on Data Mining*, pages 1146–1151. IEEE, 2011.

[35] Chang-Hwan Lee. An information-theoretic filter approach for value weighted classification learning in naive Bayes. *Data & Knowledge Engineering*, 113:116–128, 2018.

[36] H. Ma, W. Yan, Z. Yang, and H. Liu. Real-time foot-ground contact detection for inertial motion capture based on an adaptive weighted naive Bayes model. *IEEE Access*, 7: 130312–130326, 2019.

[37] Harry Zhang and Shengli Sheng. Learning weighted naive Bayes with accurate ranking. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 567–570. IEEE, 2004.

[38] Sona Taheri, John Yearwood, Musa Mammadov, and Sattar Seifollahi. Attribute weighted naive Bayes classifier using a local optimization. *Neural Computing and Applications*, 24 (5):995–1002, 2014.

[39] Diab M Diab and Khalil M El Hindi. Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification. *Applied Soft Computing*, 54: 183–199, 2017.

[40] M. Li and K. Liu. Causality-based attribute weighting via information flow and genetic algorithm for naive Bayes classifier. *IEEE Access*, 7:150630–150641, 2019.

[41] Jia Wu and Zhihua Cai. Attribute weighting via differential evolution algorithm for attribute weighted naive Bayes (WNB). *Journal of Computational Information Systems*, 7 (5):1672–1679, 2011.

[42] Liangjun Yu, Shengfeng Gan, Yu Chen, and Meizhang He. Correlation-based weight adjusted naive Bayes. *IEEE Access*, 8:51377–51387, 2020.

[43] Liangxiao Jiang, Dianhong Wang, and Zhihua Cai. Discriminatively weighted naive Bayes and its application in text classification. *International Journal on Artificial Intelligence Tools*, 21(01):1250007, 2012.

[44] Usama Fayyad and Keki Irani. Multi-interval discretization of continuous-valued attributes for classification learning. 1993.

[45] Sergio Ramírez-Gallego, Salvador García, José Manuel Benítez, and Francisco Herrera. Multivariate discretization based on evolutionary cut points selection for classification. *IEEE Transactions on Cybernetics*, 46(3):595–608, 2015.

[46] Philip Tannor and Lior Rokach. Augboost: Gradient boosting enhanced with step-wise feature augmentation. In *International Joint Conference on Artificial Intelligence*, pages 3555–3561, 2019.

[47] Bin-Bin Jia and Min-Ling Zhang. Multi-dimensional classification via kNN feature augmentation. *Pattern Recognition*, 106:107423, 2020.

[48] Huihui Li, Guihua Wen, Xiping Jia, Zhiyong Lin, Huimin Zhao, and Xiangling Xiao. Augmenting features by relative transformation for small data. *Knowledge-Based Systems*, 225:107121, 2021.

[49] Tzu-Tsung Wong. A hybrid discretization method for naïve Bayesian classifiers. *Pattern Recognition*, 45(6):2321–2325, 2012.

[50] Tong Li, Jin Li, Zheli Liu, Ping Li, and Chunfu Jia. Differentially private naive Bayes learning over multiple data sources. *Information Sciences*, 444:89–104, 2018.

[51] Chong-zhi Gao, Qiong Cheng, Pei He, Willy Susilo, and Jin Li. Privacy-preserving naive Bayes classifiers secure against the substitution-then-comparison attack. *Information Sciences*, 444:72–88, 2018.

[52] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

[53] Geoffrey I Webb, Janice R Boughton, and Zhihai Wang. Not so naive Bayes: aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

[54] L Jiang, Z Cai, and D Wang. Improving naive Bayes for classification. *International Journal of Computers and Applications*, 32(3):328–332, 2010.

[55] Liangxiao Jiang, Harry Zhang, and Zhihua Cai. A novel Bayes model: Hidden naive Bayes. *IEEE Transactions on Knowledge and Data Engineering*, 21(10):1361–1371, 2008.

[56] Ron Kohavi et al. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Knowledge Discovery and Data Mining*, volume 96, pages 202–207, 1996.

[57] Eibe Frank, Mark Hall, and Bernhard Pfahringer. Locally weighted naive Bayes. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 249–256, 2002.

[58] Liangxiao Jiang, Zhihua Cai, Harry Zhang, and Dianhong Wang. Not so greedy: Randomly selected naive Bayes. *Expert Systems with Applications*, 39(12):11022–11028, 2012.

[59] Pat Langley and Stephanie Sage. Induction of selective Bayesian classifiers. In *Uncertainty Proceedings 1994*, pages 399–406. Elsevier, 1994.

[60] Shenglei Chen, Geoffrey I Webb, Linyuan Liu, and Xin Ma. A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, 192:105361, 2020.

[61] Michael G Madden. On the classification performance of tan and general Bayesian networks. *Knowledge-Based Systems*, 22(7):489–495, 2009.

[62] Eamonn J Keogh and Michael J Pazzani. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *International Conference on Artificial Intelligence and Statistics*, 1999.

[63] Charles X Ling and Huajie Zhang. Toward Bayesian classifiers with accurate probabilities. In *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6*, pages 123–134. Springer, 2002.

[64] Liangxiao Jiang and Yuanyuan Guo. Learning lazy naive Bayesian classifiers for ranking. In *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, pages 5–pp. IEEE, 2005.

[65] Mark A Hall. Correlation-based feature selection of discrete and numeric class machine learning. 2000.

[66] François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5(9), 2004.

[67] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[68] B Venkatesh and J Anuradha. A review of feature selection and its methods. *Cybernetics and Information Technologies*, 19(1):3–26, 2019.

[69] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[70] Chotirat" ann" Ratanamahatana and Dimitrios Gunopulos. Feature selection for the naive Bayesian classifier using decision trees. *Applied Artificial Intelligence*, 17(5-6):475–487, 2003.

[71] Gunawan Herman, Bang Zhang, Yang Wang, Getian Ye, and Fang Chen. Mutual information-based method for selecting informative feature sets. *Pattern Recognition*, 46 (12):3315–3327, 2013.

[72] Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

[73] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[74] Pablo Bermejo, José A Gámez, and José M Puerta. Speeding up incremental wrapper feature subset selection with naive Bayes classifier. *Knowledge-Based Systems*, 55:140–147, 2014.

[75] Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*, pages 488–499. Springer, 2005.

[76] Sergio Ramírez-Gallego, Salvador García, Héctor Mouriño-Talín, David Martínez-Rego, Verónica Bolón-Canedo, Amparo Alonso-Betanzos, José Manuel Benítez, and Francisco Herrera. Data discretization: taxonomy and big data challenge. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(1):5–21, 2016.

[77] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[78] Chih-Fong Tsai and Yu-Chi Chen. The optimal combination of feature selection and data discretization: An empirical study. *Information Sciences*, 505:282–293, 2019.

[79] Yashuang Mu, Xiaodong Liu, Lidong Wang, and Juxiang Zhou. A parallel fuzzy rule-base based decision tree in the framework of map-reduce. *Pattern Recognition*, 103:107326, 2020.

[80] Yaling Xun, Qingxia Yin, Jifu Zhang, Haifeng Yang, and Xiaohui Cui. A novel discretization algorithm based on multi-scale and information entropy. *Applied Intelligence*, 51(2): 991–1009, 2021.

[81] Sadia Sharmin, Mohammad Shoyaib, Amin Ahsan Ali, Muhammad Asif Hossain Khan, and Oksam Chae. Simultaneous feature selection and discretization based on mutual information. *Pattern Recognition*, 91:162–174, 2019.

[82] Nuran Peker and Cemalettin Kubat. Application of chi-square discretization algorithms to ensemble classification methods. *Expert Systems with Applications*, 185:115540, 2021.

[83] Marzieh Hajizadeh Tahan and Shahrokh Asadi. Emdid: Evolutionary multi-objective discretization for imbalanced datasets. *Information Sciences*, 432:442–461, 2018.

[84] Ying Yang and Geoffrey I Webb. Proportional k-interval discretization for naive Bayes classifiers. In *European Conference on Machine Learning*, pages 564–575. Springer, 2001.

[85] Andrew KC Wong and David KY Chiu. Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6): 796–805, 1987.

[86] Philip A. Chou. Optimal partitioning for classification and regression trees. *IEEE Computer Architecture Letters*, 13(04):340–354, 1991.

[87] Jason Catlett. On changing continuous attributes into ordered discrete attributes. In *European Working Session on Learning*, pages 164–178. Springer, 1991.

[88] Randy Kerber. ChiMerge: Discretization of numeric attributes. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 123–128, 1992.

[89] Robert C Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.

[90] John Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[91] John Y. Ching, Andrew K. C. Wong, and Keith C. C. Chan. Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):641–651, 1995.

[92] Bernhard Pfahringer. Compression-based discretization of continuous attributes. In *Machine Learning Proceedings 1995*, pages 456–463. Elsevier, 1995.

[93] Xindong Wu. A Bayesian discretizer for real-valued attributes. *The Computer Journal*, 39 (8):688–691, 1996.

[94] Nir Friedman, Moises Goldszmidt, et al. Discretizing continuous attributes while learning Bayesian networks. In *International Conference on Machine Learning*, pages 157–165, 1996.

[95] Michal R Chmielewski and Jerzy W Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning*, 15(4):319–331, 1996.

[96] KM Ho and PD Scott. Zeta: a global method for discretization of cotitinuous variables. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 191–194, 1997.

[97] Jesús Cerquides and Ramon López De Màntaras. Proposal and empirical comparison of a parallelizable distance-based discretization method. In *Knowledge Discovery and Data Mining*, pages 139–142, 1997.

[98] Huan Liu and Rudy Setiono. Feature selection via discretization. *IEEE Transactions on Knowledge and Data Engineering*, 9(4):642–645, 1997.

[99] Se June Hong. Use of contextual information for feature ranking and discretization. *IEEE Transactions on Knowledge and Data Engineering*, 9(5):718–730, 1997.

[100] Djamel A Zighed, Sabine Rabaséda, and Ricco Rakotomalala. Fusinter: a method for discretization of continuous attributes. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(03):307–326, 1998.

[101] Stephen D Bay. Multivariate discretization for set mining. *Knowledge and Information Systems*, 3(4):491–512, 2001.

[102] Francis EH Tay and Lixiang Shen. A modified chi2 algorithm for discretization. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):666–670, 2002.

[103] JS Aguilar-Ruiz, JC Riquelme, FJ Ferrer-Troyano, and DS RodrÌguez-Baena. Discretization oriented to decision rules generation. *Frontiers Artificial Intelligence Application*, 82: 275–279, 2002.

[104] Marc Boulle. Khiops: A statistical discretization method of continuous attributes. *Machine Learning*, 55(1):53–69, 2004.

[105] Chao-Ton Su and Jyh-Hwa Hsu. An extended chi2 algorithm for discretization of real value attributes. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):437–441, 2005.

[106] Xiaoyan Liu and Huaiqing Wang. A discretization algorithm based on a heterogeneity criterion. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1166–1173, 2005.

[107] Sameep Mehta, Srinivasan Parthasarathy, and Hui Yang. Toward unsupervised correlation preserving discretization. *IEEE Transactions on Knowledge and Data Engineering*, 17(9): 1174–1185, 2005.

[108] Marc Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.

[109] W-H Au, Keith CC Chan, and Andrew KC Wong. A fuzzy approach to partitioning continuous attributes for classification. *IEEE Transactions on Knowledge and Data Engineering*, 18(5):715–719, 2006.

[110] Chang-Hwan Lee. A hellinger-based discretization method for numeric attributes in classification learning. *Knowledge-Based Systems*, 20(4):419–425, 2007.

[111] Qingxiang Wu, David A Bell, Girijesh Prasad, and Thomas Martin McGinnity. A distribution-index-based discretizer for decision-making with symbolic AI approaches. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):17–28, 2006.

[112] Francisco J Ruiz, Cecilio Angulo, and Núria Agell. IDD: a supervised interval distance-based method for discretization. *IEEE Transactions on Knowledge and Data Engineering*, 20(9):1230–1238, 2008.

[113] L Gonzalez-Abril, Francisco Javier Cuberos, Francisco Velasco, and Juan Antonio Ortega. Ameva: An autonomous discretization algorithm. *Expert Systems with Applications*, 36 (3):5327–5332, 2009.

[114] Ruoming Jin, Yuri Breitbart, and Chibuike Muoh. Data discretization unification. *Knowledge and Information Systems*, 19(1):1–29, 2009.

[115] Ying Yang and Geoffrey I Webb. Discretization for naive-bayes learning: managing discretization bias and variance. *Machine Learning*, 74(1):39–74, 2009.

[116] Min Li, ShaoBo Deng, Shengzhong Feng, and Jianping Fan. An effective discretization based on class-attribute coherence maximization. *Pattern Recognition Letters*, 32(15): 1962–1973, 2011.

[117] M Gethsiyal Augasta and T Kathirvalavakumar. A new discretization algorithm based on range coefficient of dispersion and skewness for neural networks classifier. *Applied Soft Computing*, 12(2):619–625, 2012.

[118] Khurram Shehzad. Edisc: a class-tailored discretization technique for rule-based classification. *IEEE Transactions on Knowledge and Data Engineering*, 24(8):1435–1447, 2011.

[119] Artur J Ferreira and Mário AT Figueiredo. An unsupervised approach to feature discretization and selection. *Pattern Recognition*, 45(9):3048–3060, 2012.

[120] Murat Kurtcephe and H Altay Güvenir. A discretization method based on maximizing the area under receiver operating characteristic curve. *International Journal of Pattern Recognition and Artificial Intelligence*, 27(01):1350002, 2013.

[121] Artur J Ferreira and Mário AT Figueiredo. Incremental filter and wrapper approaches for feature discretization. *Neurocomputing*, 123:60–74, 2014.

[122] Deqin Yan, Deshan Liu, and Yu Sang. A new approach for discretizing continuous attributes in learning systems. *Neurocomputing*, 133:507–511, 2014.

[123] Yu Sang, Heng Qi, Keqiu Li, Yingwei Jin, Deqin Yan, and Shusheng Gao. An effective discretization method for disposing high-dimensional data. *Information Sciences*, 270: 73–91, 2014.

[124] Hoang-Vu Nguyen, Emmanuel Müller, Jilles Vreeken, and Klemens Böhm. Unsupervised interaction-preserving discretization of multivariate data. *Data Mining and Knowledge Discovery*, 28(5):1366–1397, 2014.

[125] Feng Jiang and Yuefei Sui. A novel approach for discretization of continuous attributes in rough set theory. *Knowledge-Based Systems*, 73:324–334, 2015.

[126] Robert Moskovitch and Yuval Shahar. Classification-driven temporal discretization of multivariate time series. *Data Mining and Knowledge Discovery*, 29(4):871–913, 2015.

[127] Alexis Bondu, Marc Boullé, and Vincent Lemaire. A non-parametric semi-supervised discretization method. *Knowledge and Information Systems*, 24(1):35–57, 2010.

[128] Yu Zhou, Junhao Kang, Sam Kwong, Xu Wang, and Qingfu Zhang. An evolutionary multi-objective optimization framework of discretization-based feature selection for classification. *Swarm and Evolutionary Computation*, 60:100770, 2021.

[129] Binh Tran, Bing Xue, and Mengjie Zhang. A new representation in PSO for discretization-based feature selection. *IEEE Transactions on Cybernetics*, 48(6):1733–1746, 2017.

[130] Qiong Chen, Mengxing Huang, Hao Wang, and Guangquan Xu. A feature discretization method based on fuzzy rough sets for high-resolution remote sensing big data under linear spectral model. *IEEE Transactions on Fuzzy Systems*, 2021.

[131] Jesús Joel Rivas, Maria del Carmen Lara, Luis Castrejon, Jorge Hernandez-Franco, Felipe Orihuela-Espina, Lorena Palafox, Amanda Williams, Nadia Berthouze, and Enrique Sucar. Multi-label and multimodal classifier for affective states recognition in virtual rehabilitation. *IEEE Transactions on Affective Computing*, 2021.

[132] Md Geaur Rahman and Md Zahidul Islam. Discretization of continuous attributes through low frequency numerical values and attribute interdependency. *Expert Systems with Applications*, 45:410–423, 2016.

[133] Jianfeng Ren, Xudong Jiang, Junsong Yuan, and Gang Wang. Optimizing LBP structure for visual recognition using binary quadratic programming. *IEEE Signal Processing Letters*, 21(11):1346–1350, 2014.

[134] Jianfeng Ren, Xudong Jiang, and Junsong Yuan. Learning LBP structure by maximizing the conditional mutual information. *Pattern Recognition*, 48(10):3180–3190, 2015.

[135] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391. IEEE, 1995.

[136] Marzieh Hajizadeh Tahan and Shahrokh Asadi. Memod: a novel multivariate evolutionary multi-objective discretization. *Soft Computing*, 22(1):301–323, 2018.

[137] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020.

[138] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, page 109347, 2023.

[139] Feng Cen, Xiaoyu Zhao, Wuzhuang Li, and Guanghui Wang. Deep feature augmentation for occluded image classification. *Pattern Recognition*, 111:107737, 2021.

[140] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, 2021.

[141] Jin Zhang, Fuxiang Wu, Bo Wei, Qieshi Zhang, Hui Huang, Syed W Shah, and Jun Cheng. Data augmentation and dense-LSTM for human activity recognition using WiFi signal. *IEEE Internet of Things Journal*, 8(6):4628–4641, 2020.

[142] Francisco J Moreno-Barea, José M Jerez, and Leonardo Franco. Improving classification accuracy using data augmentation on small data sets. *Expert Systems with Applications*, 161:113696, 2020.

[143] Minui Hong, Jinwoo Choi, and Gunhee Kim. Stylemix: Separating content and style for enhanced data augmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14862–14870, 2021.

[144] Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and cihang xie. Shape-texture debiased neural network training. In *International Conference on Learning Representations*, 2021.

[145] Huiwen Wang, Jie Gu, and Shanshan Wang. An effective intrusion detection framework based on SVM with feature augmentation. *Knowledge-Based Systems*, 136:130–139, 2017.

[146] Wenlong Hang, Kup-Sze Choi, Shitong Wang, and Pengjiang Qian. Semi-supervised learning using hidden feature augmentation. *Applied Soft Computing*, 59:448–461, 2017.

[147] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.

[148] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39, 2022.

[149] Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)*, 52(4):1–36, 2019.

[150] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16: 321–357, 2002.

[151] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*, pages 878–887. Springer, 2005.

[152] Georgios Douzas and Fernando Bacao. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information Sciences*, 501:118–135, 2019.

[153] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[154] Xudong Jiang, Bappaditya Mandal, and Alex Kot. Eigenfeature regularization and extraction in face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):383–394, 2008.

[155] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020.

[156] Ying-Cong Chen, Xiatian Zhu, Wei-Shi Zheng, and Jian-Huang Lai. Person reidentification by camera correlation aware feature augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):392–408, 2017.

[157] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[158] Jianfeng Ren, Xudong Jiang, and Junsong Yuan. A complete and fully automated face verification system on mobile devices. *Pattern Recognition*, 46(1):45–56, 2013.

[159] C. R. Ratto, K. D. M. Jr, L. M. Collins, and P. A. Torrionfe. Bayesian context-dependent learning for anomaly classification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(4):1969–1981, 2014.

[160] Jianfeng Ren, Xudong Jiang, and Junsong Yuan. A chi-squared-transformed subspace of LBP histogram for visual recognition. *IEEE Transactions on Image Processing*, 24(6): 1893–1904, 2015.

[161] Jianfeng Ren, Xudong Jiang, Junsong Yuan, and Nadia Magnenat-Thalmann. Sound-event classification using robust texture features for robot hearing. *IEEE Transactions on Multimedia*, 19(3):447–458, 2016.

[162] Jianfeng Ren and Xudong Jiang. Regularized 2-D complex-log spectral analysis and subspace reliability analysis of micro-Doppler signature for UAV detection. *Pattern Recognition*, 69:225–237, 2017. ISSN 0031-3203.

[163] Xiaohong Wang, Xudong Jiang, and Jianfeng Ren. Blood vessel segmentation from fundus image by a cascade classification framework. *Pattern Recognition*, 88:331–341, 2019.

[164] Xudong Jiang. Linear subspace learning-based dimensionality reduction. *IEEE Signal Processing Magazine*, 28(2):16–26, 2011.

[165] Marlis Ontivero-Ortega, Agustin Lage-Castellanos, Giancarlo Valente, Rainer Goebel, and Mitchell Valdes-Sosa. Fast Gaussian naïve Bayes for searchlight classification analysis. *Neuroimage*, 163:471–479, 2017.

[166] L. Dou, X. Li, H. Ding, L. Xu, and H. Xiang. iRNA-m5C_NB: A novel predictor to identify RNA 5-methylcytosine sites based on the naive Bayes classifier. *IEEE Access*, 8: 84906–84917, 2020.

[167] P. Valdiviezo-Diaz, F. Ortega, E. Cobos, and R. Lara-Cabrera. A collaborative filtering approach based on naïve Bayes classifier. *IEEE Access*, 7:108581–108592, 2019.

[168] Yongshan Zhang, Jia Wu, Chuan Zhou, and Zhihua Cai. Instance cloned extreme learning machine. *Pattern Recognition*, 68:52–65, 2017.

[169] Xudong Jiang and Jian Lai. Sparse and dense hybrid representation via dictionary decomposition for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5):1067–1079, 2014.

[170] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation: algorithms and applications. *IEEE Access*, 3:490–530, 2015.

[171] Peng Cao, Xiaoli Liu, Jinzhu Yang, Dazhe Zhao, Min Huang, and Osmar Zaiane. L2, 1-L1 regularized nonlinear multi-task representation learning based cognitive performance prediction of Alzheimer's disease. *Pattern Recognition*, 79:195–215, 2018.

[172] Christopher M Bishop. *Pattern recognition and machine learning.* springer, 2006.

[173] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[174] Arthur Asuncion and David Newman. UCI machine learning repository, 2007.

[175] Andrew R Webb. *Statistical pattern recognition.* John Wiley & Sons, 2003.

[176] Jorge J Moré and David J Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software (TOMS)*, 20(3):286–307, 1994.

[177] Shihe Wang, Jianfeng Ren, and Ruibin Bai. A semi-supervised adaptive discriminative discretization method improving discrimination power of regularized naive Bayes. *Expert Systems with Applications*, 225:120094, 2023.

[178] Quan Ren, Hongbing Zhang, Dailu Zhang, Xiang Zhao, Lizhi Yan, Jianwen Rui, Fanxin Zeng, and Xinyi Zhu. A framework of active learning and semi-supervised learning for lithology identification based on improved naive Bayes. *Expert Systems with Applications*, 202:117278, 2022.

[179] Azka Kishwar and Adeel Zafar. Fake news detection on Pakistani news using machine learning and deep learning. *Expert Systems with Applications*, 211:118558, 2023.

[180] Lucian José Gonçales, Kleinner Farias, Lucas Silveira Kupssinskü, and Matheus Segalotto. An empirical evaluation of machine learning techniques to classify code comprehension based on EEG data. *Expert Systems with Applications*, 203:117354, 2022.

[181] Warda M Shaban, Asmaa H Rabie, Ahmed I Saleh, and MA Abo-Elsoud. Accurate detection of COVID-19 patients based on distance biased naïve Bayes (DBNB) classification strategy. *Pattern Recognition*, 119:108110, 2021.

[182] Zhiqiang Geng, Qingchao Meng, Ju Bai, Jie Chen, Yongming Han, Qin Wei, and Zhi Ouyang. A model-free Bayesian classifier. *Information Sciences*, 482:171–188, 2019.

[183] Miftahul Qorib, Timothy Oladunni, Max Denis, Esther Ososanya, and Paul Cotae. COVID-19 vaccine hesitancy: text mining, sentiment analysis and machine learning on COVID-19 vaccination twitter dataset. *Expert Systems with Applications*, 212:118715, 2023.

[184] Shihe Wang, Jianfeng Ren, Xiaoyu Lian, Ruibin Bai, and Xudong Jiang. Boosting the discriminant power of naive bayes. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 4906–4912. IEEE, 2022.

[185] Min-Ling Zhang, José M. Peña, and Victor Robles. Feature selection for multi-label naive Bayes classification. *Information Sciences*, 179(19):3218–3229, 2009. ISSN 0020-0255.

[186] Huan Zhang, Liangxiao Jiang, and Liangjun Yu. Class-specific attribute value weighting for naive Bayes. *Information Sciences*, 508:260–274, 2020.

[187] Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Exploring uncertainty in pseudo-label guided unsupervised domain adaptation. *Pattern Recognition*, 96:106996, 2019.

[188] Fereshteh Karimi, Mohammad Bagher Dowlatshahi, and Amin Hashemi. SemiACO: A semi-supervised feature selection based on ant colony optimization. *Expert Systems with Applications*, page 119130, 2022.

[189] Shengdan Hu, Duoqian Miao, and Witold Pedrycz. Multi granularity based label propagation with active learning for semi-supervised classification. *Expert Systems with Applications*, 192:116276, 2022.

[190] Jingliu Lai, Hongmei Chen, Tianrui Li, and Xiaoling Yang. Adaptive graph learning for semi-supervised feature selection with redundancy minimization. *Information Sciences*, 609:465–488, 2022.

[191] Jingliu Lai, Hongmei Chen, Weiyi Li, Tianrui Li, and Jihong Wan. Semi-supervised feature selection via adaptive structure learning and constrained graph learning. *Knowledge-Based Systems*, 251:109243, 2022.

[192] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Machine Learning Proceedings 1995*, pages 194–202. Elsevier, 1995.

[193] Wanfu Gao, Liang Hu, Ping Zhang, and Jialong He. Feature selection considering the composition of feature relevancy. *Pattern Recognition Letters*, 112:70–74, 2018.

[194] Jose Luis Flores, Borja Calvo, and Aritz Perez. Supervised non-parametric discretization based on kernel density estimation. *Pattern Recognition Letters*, 128:496–504, 2019.

[195] Jesús Alcalá-Fdez, Luciano Sanchez, Salvador Garcia, Maria Jose del Jesus, Sebastian Ventura, Josep Maria Garrell, José Otero, Cristóbal Romero, Jaume Bacardit, Victor M Rivas, et al. KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009.

[196] Shihe Wang, Jianfeng Ren, Ruibin Bai, Yuan Yao, and Xudong Jiang. A max-relevance-min-divergence criterion for data discretization with applications on naive Bayes. *Pattern Recognition*, 149:110236, 2024.

[197] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.

[198] Fei Wang, Quan Wang, Feiping Nie, Zhongheng Li, Weizhong Yu, and Fuji Ren. A linear multivariate binary decision tree classifier based on k-means splitting. *Pattern Recognition*, 107:107521, 2020.

[199] Jihong Wan, Hongmei Chen, Tianrui Li, Wei Huang, Min Li, and Chuan Luo. R2CI: Information theoretic-guided feature selection with multiple correlations. *Pattern Recognition*, page 108603, 2022.

[200] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[201] Thomas M Cover and Joy A Thomas. *Elements of Information Theory.* John Wiley & Sons, 2012.

[202] Diego MB Silva, Gustavo HA Pereira, and Tiago M Magalhães. A class of categorization methods for credit scoring models. *European Journal of Operational Research*, 296(1): 323–331, 2022.

[203] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[204] Qiushi Shi, Minghui Hu, Ponnuthurai Nagaratnam Suganthan, and Rakesh Katuwal. Weighting and pruning based ensemble deep random vector functional link network for tabular data classification. *Pattern Recognition*, 132:108879, 2022.

[205] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[206] Wa'el Hadi, Qasem A Al-Radaideh, and Samer Alhawari. Integrating associative rule-based classification with naïve Bayes for text classification. *Applied Soft Computing*, 69: 344–356, 2018.

[207] Han-joon Kim, Jiyun Kim, Jinseog Kim, and Pureum Lim. Towards perfect text classification with Wikipedia-based semantic naïve Bayes learning. *Neurocomputing*, 315:128–134, 2018.

[208] Junwu Weng, Chaoqun Weng, and Junsong Yuan. Spatio-temporal naive-Bayes nearest-neighbor (ST-NBNN) for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4171–4180, 2017.

[209] Marco Fornoni and Barbara Caputo. Scene recognition with naive Bayes non-linear learning. In *2014 22nd International Conference on Pattern Recognition*, pages 3404–3409. IEEE, 2014.

[210] Mahmood Yousefi-Azar, Vijay Varadharajan, Len Hamey, and Uday Tupakula. Autoencoder-based feature learning for cyber security applications. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3854–3861. IEEE, 2017.

[211] Jianfeng Ren, Xudong Jiang, and Junsong Yuan. Dynamic texture recognition using enhanced LBP features. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2400–2404. IEEE, 2013.

[212] Suvra Jyoti Choudhury and Nikhil R Pal. Imputation of missing data with neural networks for classification. *Knowledge-Based Systems*, 182:104838, 2019.

[213] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, 2008.

[214] Aditya Khamparia, Gurinder Saini, Babita Pandey, Shrasti Tiwari, Deepak Gupta, and Ashish Khanna. KDSAE: Chronic kidney disease classification with multimedia data learning using deep stacked autoencoder network. *Multimedia Tools and Applications*, 79(47): 35425–35440, 2020.

[215] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[216] Fu-Chen Chen and Mohammad R Jahanshahi. NB-CNN: Deep learning-based crack detection using convolutional neural network and naïve Bayes data fusion. *IEEE Transactions on Industrial Electronics*, 65(5):4392–4400, 2017.

[217] Sydney Mambwe Kasongo and Yanxia Sun. A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE Access*, 7:38597–38607, 2019.

[218] Le Hou, Vu Nguyen, Ariel B Kanevsky, Dimitris Samaras, Tahsin M Kurc, Tianhao Zhao, Rajarsi R Gupta, Yi Gao, Wenjin Chen, David Foran, et al. Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. *Pattern Recognition*, 86:188–200, 2019.

[219] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 833–840, 2011.

[220] Zhihong Zhang, Dongdong Chen, Zeli Wang, Heng Li, Lu Bai, and Edwin R Hancock. Depth-based subgraph convolutional auto-encoder for network representation learning. *Pattern Recognition*, 90:363–376, 2019.