A Bayesian Imprecise Classification method that weights instances using the error costs

Serafín Moral-García[a,*], Tahani Coolen-Maturi[b], Frank P.A. Coolen[b], Joaquín Abellán[a]

[a]Department of Computer Science and Artificial Intelligence
University of Granada, Granada, Spain
[b] Department of Mathematical Sciences
Durham University, Durham, United Kingdom

Abstract

In practical applications, Bayesian classification methods have been successfully employed. The Naïve Bayes algorithm (NB) is a quick, successful, and well-known Bayesian classification method. The Naïve Credal Classifier (NCC) is a version of NB that outputs imprecise predictions (sets of class values). NCC was also adapted for considering classification error costs. Such an adaptation is the only Bayesian method for Imprecise Classification proposed so far that considers misclassification costs. This paper presents a Bayesian algorithm for Imprecise Classification that weights the instances using the misclassification costs in such a way that the importance of an instance increases as the error cost of its class value is higher. We highlight that our proposal may provide more informative and intuitive outcomes than the existing cost-sensitive NCC. We experimentally show that our new proposed method improves the existing cost-sensitive NCC. Moreover, we highlight that our imprecise classifier has a processing time equivalent to the original NB algorithm for precise classification, which has been successfully applied to very large and real datasets. This is a crucial point in favor of our proposal because of the huge amount of data in many application areas nowadays.

*Corresponding Author
Email addresses: `seramoral@decsai.ugr.es` (Serafín Moral-García), `tahani.maturi@durham.ac.uk` (Tahani Coolen-Maturi), `frank.coolen@durham.ac.uk` (Frank P.A. Coolen), `jabellan@decsai.ugr.es` (Joaquín Abellán)

Keywords:
Bayesian methods, cost-sensitive Imprecise Classification, instance weights, Cost-Sensitive Naïve Credal Classifier, Weighted Naïve Credal Classifier

1. Introduction

Bayesian classification methods constitute a simple and effective approach to the classification task. Actually, they have been successfully used in practical applications such as *cerebral stage prediction*, [1] *traffic risk management* [2], *isotopologue detection* [3], or *android malware detection* [4]. Within Bayesian classification methods, Naïve Bayes (NB) [5] is a quick, successful, and well-known algorithm. It starts from the naïve assumption, which asserts that "all predictive features are independent given the class variable" [5]. This assumption is not always realistic. In spite of its simplicity and unrealistic assumption, the NB method has performance comparable with more complex algorithms, especially when there is no high correlation between the features [6, 7, 8]. Indeed, NB has been successfully employed in practice [9, 10, 11, 12].

Classifiers usually predict, for a given instance, a unique class state. Nevertheless, sometimes, it may be more suitable that classifiers output a set of class states as the available information is not sufficient for predicting a single one, which is known as Imprecise Classification (IC) [13]. If an instance is imprecisely classified, then a set of class values is output, constituted by those class states that are not defeated by another state via a *dominance criterion*. The obtained set of values of the class variable is called the *non-dominated states set*.

The only Bayesian method for IC proposed so far is the Naïve Credal Classifier (NCC) [13, 14], which employs the naïve assumption and the *Imprecise Dirichlet Model* (IDM) [15], a formal probability intervals model, to provide imprecise predictions. Indeed, NCC adapts the NB algorithm for IC. NCC has been successfully employed in practical applications [16, 17, 18].

Precise and imprecise classification methods tend to minimize the number of misclassified instances. This is logical when all classification errors are equally important. Nonetheless, some errors often yield higher costs than others in practical applications. For instance, in *medical diagnosis*, predicting that a patient is ill when he does not have any disease probably involves lower cost than erroneously predicting that a patient does not have any disease

[19, 20, 21, 22]; in *credit fraud detection*, the cost of a fraudulent credit card predicted as normal might be considerably higher for banks and financial institutions than the cost of a legal credit card predicted as fraudulent [23, 24, 25]; in *software defect prediction*, predicting that a non-defective module is defective may cause lower cost than predicting that a defective module is non-defective [26, 27, 28, 29].

Hence, classifiers that consider error costs, known as *cost-sensitive classifiers*, have been proposed, such as cost-sensitive Neural Networks [30, 31, 21], cost-sensitive Decision Trees [32, 33, 34], and cost-sensitive Naïve Bayes [35, 36, 37]. An evaluation measure for precise cost-sensitive classification has to take the misclassification costs into account, and an evaluation measure for cost-sensitive IC must consider the costs of erroneous classifications and the informativeness of the predictions, which is related to the cardinalities of the predicted sets of class states.

The NCC method was adapted for cost-sensitive scenarios [38]. This adaptation is called the cost-sensitive Naïve Credal Classifier (CS-NCC). As NCC, for classifying an instance, CS-NCC estimates an interval score for each class value using the prior probability interval of such a class value, the probability interval of each attribute value given that class value, and the naïve assumption. Then, CS-NCC estimates risk intervals for the class values by considering the estimated probability interval scores and the misclassification costs. Finally, CS-NCC applies a dominance criterion to such risk intervals for obtaining the set of predicted class values. So far, CS-NCC is the only Bayesian method for cost-sensitive IC.

CS-NCC employs the IDM for the estimation of the probability intervals. “This model makes previous assumptions about the data by means of a parameter. In classification, the optimal value of the IDM parameter may be different for each dataset” [39]. The Non-Parametric Predictive Inference Model (NPI-M) [40, 41] solves this drawback. This model is non-parametric and does not make prior assumptions. ”The NPI-M performs equivalently to the IDM with the best selection of the parameter in imprecise and precise classification” [42, 43].

In this research, a Bayesian method for cost-sensitive Imprecise Classification, called the *Weighted Naïve Credal Classifier* (Weighted-NCC), is developed. Our proposed algorithm computes instance weights utilizing the costs of misclassifying the class values. For classifying an instance, a probability interval score is computed for each class value using the instance weights, the NPI-M, and the naïve assumption. Then, the non-dominated states set is obtained through a dominance criterion on the computed intervals. We highlight that our proposal may lead to more intuitive and informative outputs than the existing CS-NCC.

We check the performance of Weighted-NCC over CS-NCC via an experimental study. Such a study reveals that CS-NCC precisely classifies more instances than Weighted-NCC, although the misclassification costs for precise predictions are higher with CS-NCC; for imprecise predictions, Weighted-NCC predicts fewer class states than CS-NCC even though the misclassification costs are higher with Weighted-NCC; concerning the whole performance, Weighted-NCC performs equivalently or significantly better than CS-NCC. Furthermore, it is highlighted that the processing times of CS-NCC and Weighted-NCC are equivalent to the original NB algorithm.

This paper is arranged as follows: Section 2 describes the Imprecise Classification task, the Imprecise Dirichlet Model, the Non-Parametric Predictive Inference Model, and the existing cost-sensitive Naïve Credal Classifier. Our proposed Weighted Naïve Credal Classifier is presented in Section 3. In Section 4, our experimental analysis is detailed. Section 5 concludes the paper. For ease of reference, Table 1 shows a list of symbols and notations utilized in this paper.

2. Background

2.1. Imprecise Classification

The Imprecise Classification task (IC) [13] aims to predict the set of possible class values for an instance described via a set of attributes.

Formally, IC starts from an attribute set $\{X^1, X^2, \ldots, X^n\}$ and a class variable C. Suppose that $Dom(X^i)$ is the domain of the attribute X^i, $\quad \forall i = 1, 2, \ldots, n$, and $\{c_1, c_2, \ldots, c_K\}$ are the possible values of C.

The goal of IC is to learn a function $h : Dom(X^1) \times Dom(X^2) \ldots \times Dom(X^n) \rightarrow 2^{\{c_1, c_2, \ldots, c_K\}}$ that, for a given instance described through an attribute vector $\mathbf{x}$, outputs the set of possible class values for such an instance, namely $h(\mathbf{x})$.

IC differs from standard classification, which always outcomes a unique class state. In many practical applications, IC makes more sense than standard classification because the available information may not be sufficient for making precise predictions and, thus, it might be too risky to point out a single class value.

Example 1 illustrates the importance of IC.

Table 1: List and symbols and notations employed in this work.

Symbol	Description
X	Categorical variable
$\{x_1, x_2, \ldots, x_t\}$	Set of possible values of X.
N	Number of instances.
$n(x_i)$	Number of occurrences of x_i in the sample.
s	IDM parameter.
$\mathcal{P}^{IDM}$	IDM credal set.
C	class variable.
$\{c_1, c_2, \ldots, c_K\}$	Set of class values.
$X^1, X^2, \ldots, X^n$	Set of attributes.
$\{x_1^i, x_2^1, \ldots, x_{t_i}^i\}$	Set of possible values of the attribute X^i.
$n(x^i_{r_i,k})$	Number of training instances such that $C = c_k$ and $X^i = x^i_{r_i}$.
$\mathcal{P}(C)$	Local credal set on C.
$\mathcal{P}(X^i \mid c_k)$	Credal set on X^i conditioned on $C = c_k$.
$\mathcal{P}(c_k, X^1, \ldots, X^n)$	Set of joint probability distributions.
$\underline{P}(c_k)$, $\overline{P}(c_k)$	Lower and upper probability scores on c_k.
$\underline{R}(c_k)$, $\overline{R}(c_k)$	Lower and upper risks on c_k.
$\mathcal{I}_c$	A-NPI-M probability intervals on C.
$\mathcal{P}(\mathcal{I}_c)$	A-NPI-M credal set on C.
$(\hat{p}(c_1), \hat{p}(c_2), \ldots, \hat{p}(c_K))$	Probability distribution that yields the A-NPI-M maximum entropy.
$(\hat{n}(c_1), \hat{n}(c_2), \ldots, \hat{n}(c_K))$	Arrangement that leads to the A-NPI-M maximum entropy.
m_{ij}	Cost of predicting c_i when the real class value is c_j.
$Cost(j)$	Cost corresponding to the class value c_j.
$w_1, w_2, \ldots, w_j$	Instance weights.
W_j	Sum of weights for c_j.
W	Total sum of weights.
$n_d^{WCC}(x^1_{r_1}, x^2_{r_2}, \ldots, x^K_{r_K})$	Non-dominated states set predicted by Weighted-NCC.

Example 1. *Suppose that C represents the disease that a patient can have. Let us assume that there are five possible diseases $\{c_1, c_2, c_3, c_4, c_5\}$, the disease c_1 is flu, and the disease c_2 is COVID-19. It should be noted that c_1 and c_2 could be determined by similar values of the attributes of the patient. However, for an old person with health problems, diagnosing c_1 or c_2 might lead to important differences in the death risk due to the probability of COVID given the features of the patient.*

In the previous scenario, an imprecise classifier probably outputs $\{c_1, c_2\}$, discarding the other diseases. A doctor, with this information, can make stronger tests that allow having more reliable information for deciding for one disease or the other one, or make a more hybrid treatment for the patient to improve in every situation, whenever it is possible.

If we had a precise classifier that always chooses the first class value in order (in this case, c_1), the doctor would directly carry out the corresponding treatment. This may imply a too high risk since predicting flu when *the patient has COVID-19 probably has worse consequences than predicting COVID-19 when the correct disease is flu.*

2.2. Imprecise probability models

Here, we describe two imprecise probability models for probabilistic inferences about a categorical attribute. Let X be a variable that takes values in $\{x_1, x_2, \dots, x_t\}$. Let us assume that it is disposed of N identically distributed and independent outcomes about X. Let $n(x_i)$ denote the number of occurrences of x_i in the sample, $\quad \forall i = 1, 2, \dots, t$.

2.2.1. The Imprecise Dirichlet Model

The Imprecise Dirichlet Model (IDM) [15] estimates that the probability that $X = x_i$ belongs to the following interval:

$$\mathcal{I}_i = \left[\frac{n(x_i)}{N+s}, \frac{n(x_i)+s}{N+s}\right], \quad \forall i = 1, 2, \dots, t. \tag{1}$$

where $s > 0$ is a given parameter of the model.

The probability intervals determined by Equation (1) yield the following credal set[1] on X [44]:

$$\mathcal{P}^{IDM}(X) = \{p \mid p(x_i) \in \mathcal{I}_i, \quad \forall i = 1, 2, \dots, t\}. \tag{2}$$

The parameter s is essential in the IDM. We may observe that IDM intervals are narrower as the s value is lower. In consequence, a higher value of s implies a higher imprecision degree estimated in the sample. Walley [15] suggests $s = 1$ and $s = 2$, and recommends $s = 1$.

2.2.2. Non-Parametric Predictive Inference Model

The Non-Parametric Predictive Inference Model (NPI-M) [40, 41] employs a *probability wheel* as a latent representation of the sample. On that wheel, each outcome is represented by a radial line. The wheel is partitioned into N slices equally sized. Lines corresponding to the same value must be placed next to each other. "The NPI-M is based on the circular-$A_{(N)}$ assumption, according to which the next observation falls into any slice with the same probability $\frac{1}{N}$" [41]. The value of X represented by each slice has to be

[1]A credal set is a closed and convex set of probability distributions.

decided. When two lines corresponding to the same value bound a slice, that slice has to represent such a value. When a slice is bounded by two lines associated with different values, such a slice can represent any of them, or any non-observed value.

For a given $C \subseteq \{x_1, x_2, \ldots, x_t\}$, "the NPI-M determines a probability interval in which the lower (upper) probability is computed by the minimum (maximum) fraction of slices, between all possible configurations of the wheel, that can be assigned to the values in C" [41].

Example 2. *[41] Suppose that there is an attribute called Color that takes values in*
$\{Yellow(Y), Green(G), White(W), Blue(B), Red(R), Other(O)\}$.

Suppose that we have 9 outcomes about this variable with the following frequencies:

$$n(G) = 3, \quad n(B) = 3, \quad n(Y) = 2, \quad n(R) = 1, \quad n(W) = n(O) = 0.$$

Let us assume that it is required to calculate the NPI-M probability interval of $\{B, R\}$. For the lower probability, the aim is to obtain a configuration of the wheel that assigns the minimum possible number of slices to B and R. With the available observations, it is possible to assign to B just the slices bounded by two lines that represent B, and it is not necessary that any slice represents R. In this way, Figure 1 allows us to see a configuration appropriate for the NPI-M lower probability of $\{B, R\}$. The color assigned to each slice is drawn in the slice. In such a configuration, none slice is assigned to R and just two slices correspond to B. It is easy to observe that the NPI-M lower probability of $\{B, R\}$ is equal to $\frac{2}{9}$.

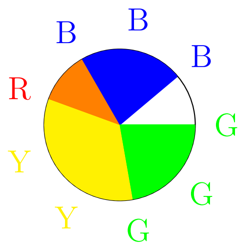

Figure 1: Configuration suitable for the lower probability of $\{B, R\}$.

For the NPI-M upper probability, we aim to obtain a configuration in which slices bounded by a line corresponding to B represent B and slices *bounded by lines representing R are assigned to R. In addition, we can separate the lines representing R and B by lines associated with another color for assigning as many slices as possible to R and B. A configuration that satisfies the mentioned points is shown in Figure 2. Hence, the NPI-M upper probability of $\{B, R\}$ is $\frac{6}{9}$.*

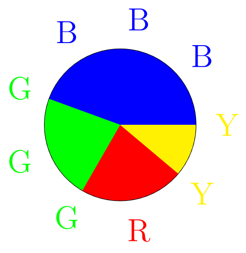

Figure 2: Configuration suitable for the upper probability of $\{B, R\}$.

As proved in [45], the NPI-M probability intervals for singletons are coherent and produce a credal set on X. In addition, for each subset of values, the NPI-M extreme probabilities can be computed from such intervals. [45]. However, "there can be probability distributions compatible with the intervals for singletons but not with the NPI-M." [45].

If all probability distributions consistent with the aforementioned intervals are considered, then an approximate model is derived. Such a model is called the Approximate Non-Parametric Predictive Inference Model (A-NPI-M). It is associated with "the convex hull of the set of probability distributions compatible with the NPI-M" [45]. Thereby, the A-NPI-M considerably simplifies the NPI-M since it evades notable constraints. In classification, the A-NPI-M and the NPI-M obtain equivalent results [42]. Considering the previous points, the A-NPI-M is employed here.

2.3. Cost-sensitive Naïve Credal Classifier

Let C denote the class variable and $\{c_1, c_2, \ldots, c_K\}$ the class values or states. Suppose that there are n attributes $X^1, X^2, \ldots, X^n$, where X^i takes values in $\left\{x_1^i, x_2^i, \ldots, x_{t_i}^i\right\}, \quad \forall i = 1, 2, \ldots, n$.

For each class value, the Naïve Credal Classifier (NCC) [13], and its adaptation for cost-sensitive scenarios (CS-NCC) [38], estimate the probability intervals in the same way.

The basis of NCC is the naïve assumption, which states that, "given the class variable, all attributes are independent" [5]:

$$\begin{aligned}
&P\left(C = c_j \mid X^1 = x^1_{r_1}, X^2 = x^2_{r_2}, \ldots, X^n = x^n_{r_n}\right) = \\
&\frac{P\left(C = c_j, X^1 = x^1_{r_1}, X^2 = x^2_{r_2}, \ldots, X^n = x^n_{r_n}\right)}{P\left(X^1 = x^1_{r_1}, X^2 = x^2_{r_2}, \ldots, X^n = x^n_{r_n}\right)} \propto \\
&P\left(C = c_j\right) P\left(X^1 = x^1_{r_1}, X^2 = x^2_{r_2}, \ldots, X^n = x^n_{r_n} \mid C = c_j\right) = \\
&P\left(C = c_j\right) \prod_{i=1}^{n} P\left(X^i = x^i_{r_i} \mid C = c_j\right), \\
&\forall r_i = 1, 2 \ldots, t_i, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, K.
\end{aligned}$$

Let us assume that there are N training instances. Let $n(c_k)$ $(n\left(x^i_{r_i}\right))$ denote the number of occurrences of c_k $(x^i_{r_i})$ in the training set,
$\forall k = 1, 2, \ldots, K, \quad r_i = 1, \ldots, t_i, \quad i = 1, 2, \ldots, n.$

NCC considers a credal set on C, namely $\mathcal{P}(C)$. Also, for each attribute X^i and class value c_k, NCC considers a credal set on X^i conditioned on $C = c_k$, $\mathcal{P}(X^i \mid c_k)$. "Such credal sets are known as *local credal sets*" [13].

Hence, for each class value c_k, a set of joint probability distributions $\mathcal{P}\left(c_k, X^1, \ldots, X^n\right)$ is derived by considering all combinations on the local credal sets and the naïve assumption:

$$\mathcal{P}\left(c_k, X^1, \ldots, X^n\right) = \left\{ p_c(c_k) \prod_{i=1}^{n} p_{ik} \mid p_c \in \mathcal{P}\left(C\right), \, p_{ik} \in \mathcal{P}\left(X^i \mid c_k\right) \right\}. \quad (3)$$

Thus, only the local credal sets are fundamental for building the NCC. In the original NCC, the IDM was used to obtain the local credal sets. In consequence, the local credal set on C is determined as follows:

$$\mathcal{P}\left(C\right) = \left\{ p \mid \frac{n\left(c_k\right)}{N+s} \leq p(c_k) \leq \frac{n\left(c_k\right) + s}{N+s}, \quad \forall k = 1, 2, \ldots, K \right\}, \quad (4)$$

s being the IDM parameter.

Likewise, for each attribute and class value, the IDM credal set on that attribute conditioned on such a class value is given by:

$$\mathcal{P}\left(X^i \mid c_k\right) = \left\{ p \mid \frac{n\left(x^i_{r_i,k}\right)}{n\left(c_k\right) + s} \leq p(x^i_{r_i} \mid c_k) \leq \frac{n\left(x^i_{r_i,k}\right) + s}{n\left(c_k\right) + s}, \quad \forall r_i = 1, \ldots, t_i \right\}, \quad (5)$$

$n\left(x^i_{r_i,k}\right)$ being the number of training instances such that $\left(C = c_k \wedge X^i = x^i_{r_i}\right)$, $\forall i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, K$.

In order to classify an instance for which $X^i = x^i_{r_i}$, with $r_i \in \{1, 2, \ldots, t_i\}$, $\forall i = 1, 2, \ldots, n$, NCC estimates probability intervals from the aforementioned credal sets.

We consider, $\forall k = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, n$:

$$\underline{p}(c_k) = \min_{p \in \mathcal{P}(C)} \{p(c_k)\}, \quad \overline{p}(c_k) = \max_{p \in \mathcal{P}(C)} \{p(c_k)\},$$

$$\underline{p}(x^i_{r_i} \mid c_k) = \min_{p_{ik} \in \mathcal{P}(X^i | c_k)} \left\{p_{ik}(x^i_{r_i} \mid c_k)\right\},$$

$$\overline{p}(x^i_{r_i} \mid c_k) = \max_{p_{ik} \in \mathcal{P}(X^i | c_k)} \left\{p_{ik}(x^i_{r_i} \mid c_k)\right\}.$$

We may note that, $\forall k = 1, 2, \ldots, K$:

$$\min_{p_c \in \mathcal{P}(C), p_{ik} \in \mathcal{P}(X^i | c_k)} \left\{ p_c(c_k) \prod_{i=1}^{n} p_{ik}(x^i_{r_i} \mid c_k) \right\} = \underline{p}(c_k) \prod_{i=1}^{n} \underline{p}(x^i_{r_i} \mid c_k),$$

$$\max_{p_c \in \mathcal{P}(C), p_{ik} \in \mathcal{P}(X^i | c_k)} \left\{ p_c(c_k) \prod_{i=1}^{n} p_{ik}(x^i_{r_i} \mid c_k) \right\} = \overline{p}(c_k) \prod_{i=1}^{n} \overline{p}(x^i_{r_i} \mid c_k).$$

Consequently, NCC estimates the following lower and upper probability scores:[2]

$$\underline{P}(c_k) = \underline{p}(c_k) \prod_{i=1}^{n} \underline{p}(x^i_{r_i} \mid c_k), \quad \overline{P}(c_k) = \overline{p}(c_k) \prod_{i=1}^{n} \overline{p}(x^i_{r_i} \mid c_k), \quad \forall k = 1, 2, \ldots, K. \tag{6}$$

Let M be the matrix of misclassification costs (K rows and K columns), in which m_{ij} denotes the cost of predicting c_i when the class state of an instance is c_j[3].

The adaptation of NCC for error costs (CS-NCC) utilizes a criterion resulting from the Bayes decision rule, according to which "the value of the

[2]These scores are the lower and upper probability values resulting from the lower and upper probability values on the local credal sets and the naïve assumption, but, indeed, they are proportional to lower and upper probability functions.

[3]The matrix of error costs is predetermined, and its values can be provided by an expert. For example, an expert in bank loans can provide the costs of not giving a loan and giving a loan that the customer does not return.

class variable with the lowest expected risk is predicted" [5], that is, the class value c_t verifying:

$$c_t = \arg \min_{i=1,2,\ldots,K} R(c_i), \tag{7}$$

where

$$R(c_i) = \sum_{j=1}^{K} m_{ij} \times P(C = c_j), \tag{8}$$

So, CS-NCC computes the lower and upper risks using the probability interval scores determined via Equation (6):

$$\underline{R}(c_i) = \sum_{j=1}^{K} m_{ij} \times \underline{P}(c_j), \quad \overline{R}(c_i) = \sum_{j=1}^{K} m_{ij} \times \overline{P}(c_j), \quad \forall i = 1, 2, \ldots, K. \tag{9}$$

On these risk intervals, CS-NCC applies a dominance criterion for obtaining the non-dominated states set. That criterion is called *strong dominance*, and establishes that "c_j dominates c_i if, and only if, $\overline{R}(c_j) \leq \underline{R}(c_i), \forall i, j \in \{1, 2, \ldots, K\}$. Strong dominance is the dominance criterion most employed in the literature" [46].

Therefore, for a given instance that satisfies $X^i = x^i_{r_i}$, with $r_i \in \{1, 2, \ldots, t_i\}$ $\forall i = 1, 2, \ldots, n$, the set of class values output by CS-NCC is given by:

$$\left\{ c_i \mid \sum_{j=1}^{K} m_{ij} \times \underline{P}(c_j) < \sum_{j=1}^{K} m_{kj} \times \overline{P}(c_j), \quad \forall k = 1, 2, \ldots, K \right\}, \tag{10}$$

where $\underline{P}(c_j)$ and $\overline{P}(c_j)$ are computed via Equation (6).

3. Weighted Naïve Credal Classifier

The Weighted Naïve Credal Classifier (Weighted-NCC), proposed here, takes the misclassification costs into account by considering instance weights. Such weights depend on the costs of incorrectly classifying the corresponding class values.

We utilize the notation of Section 2.3. The following formula is employed to compute the misclassification cost associated with the class state c_j [47]:

$$Cost(j) = \sum_{i=1}^{K} m_{ij}, \quad \forall j = 1, 2, \ldots, K. \tag{11}$$

It should be noted that the cost corresponding to c_j is obtained via the sum of the costs of predicting, for an instance with class state c_j, another class value.

We consider the A-NPI-M probability intervals on C on the training set:

$$\mathcal{I}_C = \left\{ \left[\max\left(\frac{n(c_j)-1}{N}, 0\right), \max\left(\frac{n(c_j)-1}{N}, 0\right) \right], \quad j = 1, 2, \ldots, K \right\}. \tag{12}$$

From this set of intervals, the following credal set is derived:

$$\begin{aligned} \mathcal{P}\left(\mathcal{I}_C\right) = \Bigg\{ p \mid \max\left(\frac{n(c_j)-1}{N}, 0\right) \leq p(c_j) \\ \leq \max\left(\frac{n(c_j)-1}{N}, 0\right), \quad \forall j = 1, 2, \ldots, K \Bigg\}. \end{aligned} \tag{13}$$

For quantifying the uncertainty-based information about C, the maximum entropy on $\mathcal{P}\left(\mathcal{I}_C\right)$ is a suitable uncertainty measure as it satisfies all fundamental properties [48].

Hence, we consider the arrangement of the class frequencies that yields the maximum entropy on $\mathcal{P}\left(\mathcal{I}_C\right)$, namely $(\hat{n}(c_1), \hat{n}(c_2), \ldots, \hat{n}(c_K))$. Let $(\hat{p}(c_1), \hat{p}(c_2), \ldots, \hat{p}(c_K))$ be the probability distribution for which the maximum entropy on $\mathcal{P}\left(\mathcal{I}_C\right)$ is reached. Then, $\hat{n}(c_j) = N \times \hat{p}(c_j), \quad \forall j = 1, 2, \ldots, K$. For computing that arrangement, the method proposed in [49] for the maximum entropy with the A-NPI-M is employed.

The weight of an instance with class state c_j is given by:

$$w_j = \frac{N \times Cost(j)}{\sum_{i=1}^{K} \hat{n}(c_i) \times Cost(i)}, \tag{14}$$

$Cost(j)$ being the cost of misclassifying an instance for which $C = c_j$, determined via Equation (11), $\quad \forall j = 1, 2, \ldots, K$.

Let W_j denote the sum of weights for the class state c_j: $W_j = w_j \times n(c_j), \quad \forall j = 1, 2, \ldots, K$. Let W be the total sum of weights: $W = \sum_{j=1}^{K} W_j$.

Weighted-NCC computes the following A-NPI-M probability intervals, related to the weighted training class frequencies:

$$\mathcal{I}_W = \left\{ \left[l_{w_j}, u_{w_j} \right] \right\}, \tag{15}$$

where, $\forall j = 1, 2, \ldots, K$:

$$l_{w_j} = \max\left(\frac{W_j - 1}{W}, 0\right), \quad u_{w_j} = \min\left(\frac{W_j + 1}{W}, 1\right). \tag{16}$$

The following credal set is associated with these intervals:

$$\mathcal{P}_w(C) = \left\{p \mid p(c_j) \in \left[l_{w_j}, u_{w_j}\right], \quad \forall j = 1, 2, \ldots, K\right\}. \tag{17}$$

Let us assume that we have n attributes $X^1, X^2, \ldots, X^n$ and that X^i takes values in $\left\{x_1^i, x_2^i, \ldots, x_{t_i}^i\right\}$. Suppose that we aim to classify an instance that satisfies $X^i = x_{r_i}^i$, with $r_i \in \{1, 2, \ldots, t_i\} \quad \forall i = 1, 2, \ldots, n$.

Weighted-NCC considers the local credal set on C given by Equation (17). Concerning the local credal sets corresponding to the attributes conditioned on the class values, it considers the sum of weights of the training instances that have an attribute value and a certain value of the class variable:

$$W_{x_{r_i,j}^i} = n\left(x_{r_i,j}^i\right) \times w_j, \tag{18}$$

$n\left(x_{r_i,j}^i\right)$ being the number of training instances such that $C = c_j$ and $X^i = x_{r_i}^i, \quad \forall i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, K$.

Let us denote:

$$l_{x_{r_i}^i, w_j} = \max\left(\frac{W_{x_{r_i,j}^i} - 1}{W_j}, 0\right), \quad u_{x_{r_i}^i, w_j} = \min\left(\frac{W_{x_{r_i,j}^i} + 1}{W_j}, 1\right), \tag{19}$$
$$\forall i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, K.$$

Weighted-NCC considers the following credal set on X^i conditioned on $C = c_j$:

$$\mathcal{P}_{w_j, X^i}(X^i \mid c_j) = \left\{p \mid p(x_{r_i}^i) \in \left[l_{x_{r_i}^i, w_j}, u_{x_{r_i}^i, w_j}\right], \quad \forall i = 1, 2, \ldots, n\right\}. \tag{20}$$

Hence, Weighted-NCC considers, for each class value c_j, the set of probability distributions that derives from the naïve assumption and from making all possible combinations on the local credal set on C, defined in Equation (17), and the local credal sets on the attributes conditioned on the class values, defined in Equation (20):

$$\mathcal{P}_{w_j}\left(c_j, X^1, \ldots, X^n\right) = \left\{p_c(c_j) \prod_{i=1}^{n} p_{ij} \mid p_c \in \mathcal{P}_w\left(C\right), \, p_{ij} \in \mathcal{P}_{w_j, X^i}\left(X^i \mid c_j\right)\right\}. \tag{21}$$

We may deduce that, $\forall j = 1, 2, \ldots, K$:

$$\min_{p_c \in \mathcal{P}_w(C),\, p_{ij} \in \mathcal{P}_{w_j, X^i}(X^i|c_j)} \left\{ p_c(c_j) \prod_{i=1}^{n} p_{ij}(x^i_{r_i} \mid c_j) \right\} = l_{w_j} \times \prod_{i=1}^{n} l_{x^i_{r_i}, w_j},$$
$$\max_{p_c \in \mathcal{P}_w(C),\, p_{ij} \in \mathcal{P}_{w_j, X^i}(X^i|c_j)} \left\{ p_c(c_j) \prod_{i=1}^{n} p_{ij}(x^i_{r_i} \mid c_j) \right\} = u_{w_j} \times \prod_{i=1}^{n} u_{x^i_{r_i}, w_j}.$$

In consequence, the probability interval scores computed by Weighted-NCC for each class state are given by:

$$\underline{P}_{w_j}(c_j) = l_{w_j} \times \prod_{i=1}^{n} l_{x^i_{r_i}, w_j}, \quad \overline{P}_{w_j}(c_j) = u_{w_j} \times \prod_{i=1}^{n} u_{x^i_{r_i}, w_j}, \quad \forall j = 1, 2, \ldots, K. \tag{22}$$

The *stochastic dominance* criterion is applied on these intervals for obtaining the non-dominated states set. It is the strongest dominance criterion on probability intervals and the most employed in these situations [46]. The stochastic dominance criterion states that "c_j dominates c_k if, and only if, $\underline{P}_{w_j}(c_j) \geq \overline{P}_{w_k}(c_k)$" [46], $\quad \forall j, k = 1, 2, \ldots, K$.

Thus, for an instance that satisfies $X^i = x^i_{r_i}, \quad \forall i = 1, 2, \ldots, n$, the set of class values predicted by Weighted-NCC is determined as follows:

$$n_d^{WCC}(x^1_{r_1}, x^2_{r_2}, \ldots, x^n_{r_n}) = \left\{ c_k \mid u_{w_k} \times \prod_{i=1}^{n} u_{x^i_{r_i}, w_j} > l_{w_j} \times \prod_{i=1}^{n} l_{x^i_{r_i}, w_j}, \quad \forall j = 1, 2, \ldots, K \right\}, \tag{23}$$

where l_{w_j} and u_{w_j} are given by Equation (16), and $l_{x^i_{r_i}, w_j}$ and $u_{x^i_{r_i}, w_j}$ by Equation (19), $\quad \forall i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, K$.

Algorithm 1 summarizes Weighted-NCC.

3.1. Time complexity of Weighted-NCC

Our procedure needs to obtain an interval with extreme values similar to the values used in the NB classifier. We also need to use a matrix about the cost of errors with dimension $K \times K$, K being the number of class values.

The extreme values of the intervals have the same complexity of calculus as the ones corresponding to the NB classifier. NB has the computational complexity $O(nK)$, where n is the number of predictive attributes

```
Procedure **Weighted-NCC**(training class frequencies
$(n(c_1), n(c_2), \ldots, n(c_K))$, attribute frequencies
$n\left(x^i_{r_i,j}\right), \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, K)$, instance to classify
such that $X^i = x^i_{r_i}$
Compute the instance weights via Equation (14)
Compute the A-NPI-M probability intervals associated with the
  weighted training class frequencies (Equation (16)) and its
  corresponding credal set, $\mathcal{P}_w(C)$ (Equation (17))
for $i = 1$ to $n$ do
    for $j = 1$ to $K$ do
        Consider the credal set on $X^i$ conditioned on $C = c_j$,
          $\mathcal{P}_{w_j, X^i}(X^i \mid c_j)$, computed through Equation (20)
    end
end
for $j = 1$ to $K$ do
    Compute the set of joint probability distributions
      $\mathcal{P}_{w_j}(c_j, X^1, \ldots, X^n)$ via Equation (21)
end
for $j = 1$ to $K$ do
    Compute the lower and upper scores for $c_j$, $\underline{P}_{w_j}(c_j)$ and $\overline{P}_{w_j}(c_j)$,
      by means of Equation (22)
end
Determine the non-dominated states set, $n_d^{WCC}(x^1_{r_1}, x^2_{r_2}, \ldots, x^n_{r_n})$, via
  Equation (23)
return $n_d^{WCC}(x^1_{r_1}, x^2_{r_2}, \ldots, x^n_{r_n})$
```
Algorithm 1: Summary of our proposed Weighted-NCC method.

[50]. Hence, our procedure has a computational complexity of $O(nK^2)$. We must remark that the $K \times K$ values of the matrix of error costs constitute a set of fixed (constant) and previously known values.

NB is a very quick procedure that can be used in very large real datasets. Due to the similar complexity of our algorithm, it can be stated that our new imprecise procedure can be used in a similar way. The experiments related to processing time, carried out in Section 4.3, corroborate this issue.

3.2. Why our proposed cost-sensitive NCC?

The main differences of our proposed Weighted-NCC versus the existing CS-NCC can be summarized as follows:

- The existing CS-NCC uses the IDM to compute the probability intervals for the class variable and the probability intervals of the attribute values given the class values. In contrast, Weighted-NCC employs the A-NPI-M to compute such extreme probabilities. As explained before, the A-NPI-M is more suitable than the IDM because it does not require the choice of a model parameter.

- Our proposed Weighted-NCC considers a weight for each training instance based on the cost of incorrectly classifying its class value. Thereby, for estimating the extreme probabilities, the larger the misclassification cost of a class state, the higher the importance of an instance with that class state.

  Nonetheless, CS-NCC computes all lower and upper probability scores by assuming the same importance for all instances. Then, for each class state, CS-NCC estimates a risk interval based on the costs of outputting such a class value and the probability interval scores of the other class values. Consequently, the interval probability score estimated for that class state is not considered for computing the corresponding risk interval. The same happens with the misclassification cost associated with that class value.

  Hence, we could say that the outcomes of Weighted-NCC may be more informative than the ones of CS-NCC. Example 3 illustrates this issue.

- Also, in some cases, CS-NCC makes precise predictions when it might be more intuitive to predict more than one class value. This point is shown in Example 4, where this drawback is mitigated with our developed Weighted-NCC.

Example 3. *Let us assume that there are four possible values of the class variable* $\{c_1, c_2, c_3, c_4\}$*. Suppose that there are* $N = 400$ *training instances and that* $n(c_j) = 100$*, for* $j = 1, 2, 3, 4$*. Let* M *be the matrix of misclassification costs with the following values:*

$$\begin{aligned} m_{jj} &= 0, \quad \forall j = 1, 2, 3, 4, \\ m_{ij} &= j, \quad \forall i, j \in \{1, 2, 3, 4\}, \quad j \neq i. \end{aligned}$$

We have the following misclassification costs for the class values (Equation 11):

$$\begin{aligned} Cost(1) &= m_{21} + m_{31} + m_{41} = 3, \\ Cost(2) &= m_{12} + m_{32} + m_{42} = 6, \\ Cost(3) &= m_{13} + m_{23} + m_{43} = 9, \\ Cost(4) &= m_{14} + m_{24} + m_{34} = 12. \end{aligned}$$

We may note that, in this situation, the given arrangement of the class frequencies, associated with the uniform probability distribution, coincides with the one that gives rise to the maximum entropy with the A-NPI-M. Therefore, we have the following instance weights (Equation (14)):

$$w_1 = 0.4, \quad w_2 = 0.8, \quad w_3 = 1.2, \quad w_4 = 1.6.$$

Let X^1 *and* X^2 *be two attributes. Suppose that we want to classify an instance for which* $X^1 = x_0^1$ *and* $X^2 = x_0^2$*. For these attribute values, let us assume the following observed training class frequencies:*

$$\begin{aligned} X^1 = x_0^1 \rightarrow \quad & n(c_1) = 21, \quad n(c_2) = 19, \quad n(c_3) = n(c_4) = 15. \\ X^2 = x_0^2 \rightarrow \quad & n(c_1) = 11, \quad n(c_2) = 8, \quad n(c_3) = n(c_4) = 4. \end{aligned}$$

With regard to the IDM parameter, let us assume the standard value $s = 1$*. For CS-NCC, we have the following lower and upper probability scores, computed by means of Equation (6):*

$$\begin{aligned} \underline{P}(c_1) = 0.0056, \quad & \underline{P}(c_2) = 0.0037, \quad \underline{P}(c_3) = \underline{P}(c_4) = 0.0015, \\ \overline{P}(c_1) = 0.0065, \quad & \overline{P}(c_2) = 0.0044, \quad \overline{P}(c_3) = \overline{P}(c_4) = 0.002. \end{aligned}$$

The risk intervals are obtained by:

$$\begin{aligned}
\underline{R}(c_1) &= \underline{P}(c_2)m_{12} + \underline{P}(c_3)m_{13} + \underline{P}(c_4)m_{14} = 0.0122,\\
\overline{R}(c_1) &= \overline{P}(c_2)m_{12} + \overline{P}(c_3)m_{13} + \overline{P}(c_4)m_{14} = 0.0156,\\
\underline{R}(c_2) &= \underline{P}(c_1)m_{21} + \underline{P}(c_3)m_{23} + \underline{P}(c_4)m_{24} = 0.0123,\\
\overline{R}(c_2) &= \overline{P}(c_1)m_{21} + \overline{P}(c_3)m_{23} + \overline{P}(c_4)m_{24} = 0.0154,\\
\underline{R}(c_3) &= \underline{P}(c_1)m_{31} + \underline{P}(c_2)m_{32} + \underline{P}(c_4)m_{34} = 0.0143,\\
\overline{R}(c_3) &= \overline{P}(c_1)m_{31} + \overline{P}(c_2)m_{32} + \overline{P}(c_4)m_{34} = 0.0169,\\
\underline{R}(c_4) &= \underline{P}(c_1)m_{41} + \underline{P}(c_2)m_{42} + \underline{P}(c_3)m_{43} = 0.0142,\\
\overline{R}(c_4) &= \overline{P}(c_1)m_{41} + \overline{P}(c_2)m_{42} + \overline{P}(c_3)m_{43} = 0.0162.
\end{aligned}$$

In this way, $\underline{R}(c_j) < \overline{R}(c_k), \quad \forall j,k = 1,2,3,4$. *Therefore, according to the stochastic dominance criterion on these intervals, all class values are non-dominated.*

Concerning the lower and upper probability scores estimated by our proposed Weighted-NCC method, we have that:

$$\underline{P}_{w_j}(c_j) = l_{w_j} \times l_{x_0^1,w_j} \times l_{x_0^2,w_j}, \quad \overline{P}_{w_j}(c_j) = u_{w_j} \times u_{x_0^1,w_j} \times u_{x_0^2,w_j},$$

where l_{w_j} *and* u_{w_j} *are determined by Equation (16) and* $l_{x_0^1,w_j}$, $l_{x_1^1,w_j}$, $u_{x_0^1,w_j}$, *and* $u_{x_0^2,w_j}$ *by Equation (19),* $\quad \forall j = 1,2,3,4$.

In consequence, we have the following lower and upper probability scores:

$$\begin{aligned}
&\underline{P}_{w_1}(c_1) = 0.0015, \quad \underline{P}_{w_2}(c_2) = 0.0024, \quad \underline{P}_{w_3}(c_3) = 0.0013, \quad \underline{P}_{w_4}(c_4) = 0.0019,\\
&\overline{P}_{w_1}(c_1) = 0.0033, \quad \overline{P}_{w_2}(c_2) = 0.0038, \quad \overline{P}_{w_3}(c_3) = 0.0023, \quad \overline{P}_{w_4}(c_4) = 0.0029.
\end{aligned}$$

This implies that, under the stochastic dominance criterion on these intervals, c_3 *is dominated by* c_2 *because* $\underline{P}_{w_2}(c_2) > \overline{P}_{w_3}(c_3)$.

In this scenario, the prediction of Weighted-NCC is probably more intuitive than the prediction of CS-NCC since the conditional frequencies of c_3 *are far lower than the conditional frequencies of* c_2*, and the misclassification cost of* c_3 *is not much higher than the one of* c_2.

Example 4. *Suppose that we have three class values* $\{c_1, c_2, c_3\}$ *and* $N = 300$ *training instances. Let us assume that* $n(c_j) = 100$*, for* $j = 1,2,3$. *Suppose that the matrix* M *of error costs is given by:*

$$\begin{aligned}
m_{jj} &= 0, \quad \forall j = 1,2,3,\\
m_{ij} &= j, \quad \forall i,j \in \{1,2,3\}, \quad j \neq i.
\end{aligned}$$

We have the following misclassification costs:

$$\begin{aligned} Cost(1) &= m_{21} + m_{31} = 2, \\ Cost(2) &= m_{12} + m_{32} = 4, \\ Cost(3) &= m_{13} + m_{23} = 6. \end{aligned}$$

As in Example 3, the given arrangement of the class frequencies coincides with the one that leads to the maximum entropy with the A-NPI-M. Hence, we have the following instance weights (Equation (14)):

$$w_1 = 0.5, \quad w_2 = 1, \quad w_3 = 1.5.$$

Let X^1 and X^2 be two attributes. Suppose that we want to classify an instance for which $X^1 = x_0^1$ and $X^2 = x_0^2$. Let us assume the following observed training class frequencies for these attribute values:

$$\begin{aligned} X^1 = x_0^1 \rightarrow \quad & n(c_1) = 95, \quad n(c_2) = 81, \quad n(c_3) = 68. \\ X^2 = x_0^2 \rightarrow \quad & n(c_1) = 95, \quad n(c_2) = 81, \quad n(c_3) = 68. \end{aligned}$$

Assuming the value $s = 1$ for the IDM parameter, we have the following lower and upper probability scores estimated by CS-NCC (Equation 6):

$$\begin{aligned} \underline{P}(c_1) = 0.0056, \quad & \underline{P}(c_2) = 0.2127 \quad \underline{P}(c_3) = 0.1506, \\ \overline{P}(c_1) = 0.0065, \quad & \overline{P}(c_2) = 0.2212 \quad \overline{P}(c_3) = 0.1566. \end{aligned}$$

The risk intervals are obtained by:

$$\begin{aligned} \underline{R}(c_1) &= \underline{P}(c_2)m_{12} + \underline{P}(c_3)m_{13} = 0.8791, \\ \overline{R}(c_1) &= \overline{P}(c_2)m_{12} + \overline{P}(c_3)m_{13} = 0.9122, \\ \underline{R}(c_2) &= \underline{P}(c_1)m_{21} + \underline{P}(c_3)m_{23} = 0.7457, \\ \overline{R}(c_2) &= \overline{P}(c_1)m_{21} + \overline{P}(c_3)m_{23} = 0.773, \\ \underline{R}(c_3) &= \underline{P}(c_1)m_{31} + \underline{P}(c_2)m_{32} = 0.7212, \\ \overline{R}(c_3) &= \overline{P}(c_1)m_{31} + \overline{P}(c_2)m_{32} = 0.7455. \end{aligned}$$

According to the stochastic dominance criterion, both c_1 and c_2 are dominated by c_3 as $\overline{R}(c_3) < \underline{R}(c_j)$, for $j = 1, 2$.

The following lower and upper probability scores are estimated by our proposed Weighted-NCC:

$$\underline{P}_{w_j}(c_j) = l_{w_j} \times l_{x_0^1, w_j} \times l_{x_0^2, w_j}, \quad \overline{P}_{w_j}(c_j) = u_{w_j} \times u_{x_0^1, w_j} \times l_{x_0^2, w_j},$$

where l_{w_j} and u_{w_j} are given by Equation (16) and $l_{x_0^1,w_j}$, $l_{x_1^1,w_j}$, $u_{x_0^1,w_j}$, and $u_{x_0^2,w_j}$ by Equation (19), for $j = 1, 2, 3$.

Hence,

$$\begin{aligned}
&\underline{P}_{w_1}(c_1) = 0.1413, \quad &&\underline{P}_{w_2}(c_2) = 0.2112, \quad &&\underline{P}_{w_3}(c_3) = 0.2252,\\
&\overline{P}_{w_1}(c_1) = 0.1600, \quad &&\overline{P}_{w_2}(c_2) = 0.2264, \quad &&\overline{P}_{w_3}(c_3) = 0.2373.
\end{aligned}$$

In this case, c_1 is dominated by both c_2 and c_3 since $\underline{P}_{w_j}(c_j) > \overline{P}_{w_1}(c_1)$, for $j = 2, 3$. However, c_2 and c_3 are non-dominated because $\overline{P}_{w_j}(c_2) > \underline{P}_{w_1}(c_3)$ and $\overline{P}_{w_j}(c_3) > \underline{P}_{w_1}(c_2)$.

It makes sense that c_3 dominates c_1 since the misclassification cost of c_3 is much greater than the one of c_1 and the conditional frequencies of c_3 are not drastically lower than the ones of c_1. The misclassification cost of c_2 is lower than the one of c_3 but the conditional frequencies of c_2 are higher than the ones of c_3. Intuitively, the trade-off between conditional frequencies and misclassification costs of c_2 and c_3 are equivalent. Therefore, we can state that, in this situation, the prediction made by our proposal might be more intuitive than the one made by CS-NCC.

To sum up, the predictions made by the proposed Weighted-NCC might be more intuitive and informative than the ones made by the existing CS-NCC. In consequence, Weighted-NCC may perform better than CS-NCC. We validate this via experimentation in Section 4.

4. Experiments

4.1. Experimental settings

4.1.1. Datasets

We have used 34 datasets from *UCI Machine Learning Repository* [51] to check the performance of the two methods considered in our experimental analysis. These datasets coincide with the ones employed in the experimental studies carried out in [38, 52] for comparing cost-sensitive Imprecise Classification algorithms. They are varied concerning size, number of class values, number of features (continuous and categorical), and ranges of values for discrete variables. We have only chosen datasets that have three or more class states since, “with only two class states, an Imprecise Classification method always predicts all values of the class variable or a unique one” [38]. The essential characteristics of each dataset are presented in Table 2.

Table 2: Datasets used in this experimental study. “N” indicates the number of instances, “Attr” is the number of attributes, “Cont” and “Disc” mean, respectively, the number of continuous and discrete attributes, “K” is the number of class states, and “Range” indicates the range of values of the discrete variables.

Dataset	N	Attr	Cont	Disc	K	Range
anneal	898	38	6	32	6	2-10
arrhythmia	452	279	206	73	16	2
audiology	226	69	0	69	24	2-6
autos	205	25	15	10	7	2-22
balance-scale	625	4	4	0	3	-
car	1728	6	0	6	4	3-4
cmc	1473	9	2	7	3	2-4
dermatology	366	34	1	33	6	2-4
ecoli	366	7	7	0	7	-
flags	194	30	2	28	8	2-13
hypothyroid	3772	30	7	23	4	2-4
iris	150	4	4	0	3	-
letter	20000	16	16	0	26	-
lymphography	146	18	3	15	4	2-8
mfeat-pixel	2000	240	0	240	10	4-6
nursery	12960	8	0	8	4	2-4
optdigits	5620	64	64	0	10	-
page-blocks	5473	10	10	0	5	-
pendigits	10992	16	16	0	10	-
postop-patient-data	90	9	0	9	3	2-4
primary-tumor	339	17	0	17	21	2-3
segment	2310	19	16	0	7	-
soybean	683	35	0	35	19	2-7
spectrometer	531	101	100	1	48	4
splice	3190	60	0	60	3	4-6
sponge	76	44	0	44	3	2-9
tae	151	5	3	2	3	2
vehicle	946	18	18	0	4	-
vowel	990	11	10	1	11	2
waveform	5000	40	40	0	3	-
wine	178	13	13	0	3	-
zoo	101	16	1	16	7	2

4.1.2. Procedure

We have applied the preprocessing used in the experimental comparisons between IC methods carried out in [38, 53, 54] to the datasets considered in this experimentation: we have replaced missing values with modal values for discrete variables and with mean values for continuous attributes. Afterward, we have discretized the datasets by using Fayyad and Irani's discretization algorithm [55]. For the preprocessing, we have employed the filters for discretization and missing values available in the Weka software [56].

In this experimental study, we have used two methods: CS-NCC and Weighted-NCC, the only existing Bayesian algorithms for cost-sensitive IC. Both algorithms have been implemented in Weka. Consistently with the experimental analyses carried out in [38, 52], we have utilized five cost matrices, which are described below.

- "**Cost Matrix 0/1**: All incorrect predictions have a cost equal to 1" [38].
- "**Cost Matrix (I)**: Only the real class values influence the cost of incorrect predictions. The lower is the frequency of a class value, the higher is the cost associated with it" [38].
- "**Cost Matrix (II)**: The cost of an erroneous classification is just influenced by the predicted class value. The more observed values of the class variable have less cost than the ones less observed" [38].
- "**Cost Matrix (III)**: It is similar to Cost Matrix (I). However, now the higher the frequency of a class value, the higher its corresponding cost" [38].
- "**Cost Matrix (IV)**: This cost matrix is equivalent to Cost Matrix (II), but now the more observed values of the class variable have higher costs than the less observed class values" [38].

We have repeated a 10-fold cross-validation procedure 10 times for each cost matrix and preprocessed dataset.

4.1.3. Evaluation metrics

As said in the Introduction, an evaluation metric for cost-sensitive Imprecise Classification must consider the informativeness of the predictions and the misclassification costs. The following two metrics evaluate how informative an imprecise classifier is:

- **Determinacy**: It is the fraction of instances precisely classified.
- **Indeterminacy Size**: It indicates, among the instances such that two or more class values are output, the average number of non-dominated states.

Concerning the misclassification costs, the two following evaluation measures are useful [52]:

- **Single Cost**: It indicates the average cost of erroneous classifications between the instances precisely classified.
- **Set Cost**: "It consists of the average cost of misclassifications among the instances imprecisely classified. The cost of an incorrect imprecise classification is determined by the maximum cost of predicting a non-dominated class value" [13].

  Formally, let c_{t_i} be the true value of the class variable of the i-th test instance, where $t_i \in \{1, 2, \ldots, K\}$, and U_i the predicted set of class values for that instance. If the i-th test instance is misclassified, then the following value is considered:

  $$\alpha_{t_i} = \max_{c_j \in U_i} m_{jt_i}. \tag{24}$$

  Set Cost is determined by:

  $$\frac{1}{|\{1 \leq i \leq N_{test} : |U_i| \geq 2\}|} \times \sum_{i=1, |U_i| \geq 2 \wedge i:Error}^{N_{test}} \alpha_{t_i}, \tag{25}$$

  N_{test} being the number of test instances.

To measure the whole performance of the two algorithms considered here, the MIC evaluation metric, introduced by Abellán and Masegosa [38], has been employed because it is the well-established to test the performance of a cost-sensitive imprecise classifier. Such an evaluation metric is defined as:

$$MIC = \frac{1}{N_{test}} \times \left(- \sum_{i:Correct} \log_2 \frac{|U_i|}{K} - \frac{1}{K-1} \times \sum_{i:Error} -\alpha_{t_i} \times \log_2 K \right). \tag{26}$$

The MIC evaluation metric strictly penalizes the errors. Its optimal value, $\log_2(K)$, is attained when a unique class value is correctly predicted for all instances. If all class values are always predicted by a cost-sensitive imprecise classifier, then the MIC value is equal to 0. This is logical because, in that scenario, the classifier is non-informative.

4.1.4. Statistical comparisons

Considering the indications given in [57] for statistical comparisons between two algorithms, we compare, for each cost matrix, the results of CS-NCC and Weighted-NCC in all the evaluation metrics considered here by means of the following tests:

- **Corrected Paired t-test**: It is employed for comparing two algorithms applied to a single dataset. This test checks whether one method outperforms the other one across multiple runs of a cross-validation procedure on a dataset.
- **Wilcoxon test** [58]: This test considers, for each dataset, the normalized difference among the results of the two algorithms, regardless of signs. Then, it ranks the differences and compares negative and positive ranks.

A significance level of $\alpha = 0.05$ has been utilized for the aforementioned tests.

4.2. Results and discussion

Tables 3, 4,5, and 6 summarize the results of each method for each cost matrix in Single Cost, Set Cost Determinacy, and Indeterminacy Size, respectively. Specifically, these tables show, for each metric and cost matrix, the average results, in how many datasets a method significantly outperforms the other one via the Corrected Paired t-test, and whether an algorithm achieves a significantly better performance than the other one according to the Wilcoxon test. We mark in bold the best results obtained in Average and Corrected Paired t-test. Also, Figures 3, 4, 5, and 6 illustrate the average results obtained by each classifier for each cost matrix in Single Cost, Set Cost, Determinacy, and Indeterminacy Size, respectively.

Concerning these results, we can remark the following points:

Table 3: Summary of Single Cost results for each cost matrix. In the "Wilcoxon test" rows, "*" means that the method of the column significantly outperforms the other one through the Wilcoxon test for the corresponding cost matrix. The rows "Paired t-test" show the number of datasets in which the algorithm of the column performs significantly better than the other one via the Corrected Paired t-test for the corresponding cost matrix.

		CS-NCC	Weighted-NCC
Cost Matrix 0/1:	Wilcoxon test		*
	Paired t-test	0	**13**
	Average	0.0689	**0.06**
Cost Matrix (I):	Wilcoxon test	=	=
	Paired t-test	5	**10**
	Average	0.2434	**0.2316**
Cost Matrix (II):	Wilcoxon test	=	=
	Paired t-test	7	7
	Average	**0.1655**	0.2274
Cost Matrix (III):	Wilcoxon test		*
	Paired t-test	0	**17**
	Average	0.3156	**0.2198**
Cost Matrix (IV):	Wilcoxon test	*	
	Paired t-test	**10**	1
	Average	**0.1761**	0.2613

- Regarding **Single Cost** (Table 3), Weighted-NCC significantly outperforms CS-NCC via the Wilcoxon test for Cost Matrices 0/1 and (III), while CS-NCC achieves a significantly better performance than Weighted-NCC for Cost Matrix (IV). For Cost Matrices (I) and (II), Weighted-NCC and CS-NCC perform equivalently according to the Wilcoxon test in Single Cost. Weighted-NCC achieves a lower average Single Cost value than CS-NCC for Cost Matrices 0/1, (I), and (III), while, for Cost Matrices (II) and (IV), the opposite happens (See Figure 3). According to the Corrected Paired t-test, for Cost Matrices 0/1, (I), and (III), Weighted-NCC performs significantly better in more datasets than CS-NCC. The contrary occurs for Cost Matrix (IV). Consequently, for instances precisely classified, the misclassification costs of Weighted-NCC are generally lower than the misclassifications costs of CS-NCC, although it depends on the matrix of error costs considered.

- Due to the results achieved in **Set Cost** (Table 4), it can be stated that, for instances imprecisely classified, the misclassification costs of CS-NCC are much lower than the misclassification costs of Weighted-

Table 4: Summary of Set Cost results for each cost matrix. In the "Wilcoxon test" rows, "*" means that the method of the column significantly outperforms the other one through the Wilcoxon test for the corresponding cost matrix. The rows "Paired t-test" show the number of datasets in which the algorithm of the column performs significantly better than the other one via the Corrected Paired t-test for the corresponding cost matrix.

		CS-NCC	Weighted-NCC
Cost Matrix 0/1:	Wilcoxon test	*	
	Paired t-test	**17**	0
	Average	**0.0094**	0.0291
Cost Matrix (I):	Wilcoxon test	*	
	Paired t-test	**15**	0
	Average	**0.0201**	0.0923
Cost Matrix (II):	Wilcoxon test	*	
	Paired t-test	14	3
	Average	**0.0629**	0.2876
Cost Matrix (III):	Wilcoxon test	*	
	Paired t-test	**19**	1
	Average	**0.0265**	0.2831
Cost Matrix (IV):	Wilcoxon test	=	=
	Paired t-test	**10**	3
	Average	**0.2265**	0.2425

NCC. Indeed, for the five cost matrices, CS-NCC gets a lower average Set Cost value than Weighted-NCC (See Figure 4), and the number of datasets where CS-NCC achieves a significantly better performance than Weighted-NCC in Set Cost according to the Corrected Paired t-test is much greater than the number of datasets where the contrary happens. Furthermore, CS-NCC performs significantly better than Weighted-NCC via the Wilcoxon test in Set Cost for all cost matrices, except for Cost Matrix (IV).

- **Determinacy** results (Table 5) reveal that the existing CS-NCC precisely classifies more instances than our proposed Weighted-NCC. In fact, for the five cost matrices, the average Determinacy value achieved by CS-NCC is higher than the one obtained by Weighted-NCC (See Figure 5), and the number of datasets in which CS-NCC significantly outperforms Weighted-NCC in Determinacy according to the Corrected Paired t-test is much higher than the number of datasets in which the contrary occurs. Moreover, for Cost Matrices 0/1, (II), and (III), CS-NCC achieves a significantly better performance than Weighted-NCC

Table 5: Summary of Determinacy results for each cost matrix. In the "Wilcoxon test" rows, "*" means that the method of the column significantly outperforms the other one through the Wilcoxon test for the corresponding cost matrix. The rows "Paired t-test" show the number of datasets in which the algorithm of the column performs significantly better than the other one via the Corrected Paired t-test for the corresponding cost matrix.

		CS-NCC	Weighted-NCC
Cost Matrix 0/1:	Wilcoxon test	*	
	Paired t-test	**20**	0
	Average	**0.5781**	0.5108
Cost Matrix (I):	Wilcoxon test	=	=
	Paired t-test	**15**	6
	Average	**0.5853**	0.5713
Cost Matrix (II):	Wilcoxon test	*	
	Paired t-test	**16**	2
	Average	**0.6096**	0.5212
Cost Matrix (III):	Wilcoxon test	*	
	Paired t-test	**27**	0
	Average	**0.6159**	0.441
Cost Matrix (IV):	Wilcoxon test	=	=
	Paired t-test	**7**	5
	Average	**0.5762**	0.559

in Determinacy according to the Wilcoxon test.

- Concerning **Indeterminacy Size** (Table 6), Weighted-NCC significantly outperforms CS-NCC via the Wilcoxon test for all cost matrices. Also, for the five cost matrices, the average Indeterminacy Size value attained by Weighted-NCC is considerably lower than the one obtained by CS-NCC (See Figure 6), and the results of the Corrected Paired t-test in Indeterminacy Size indicate that the number of datasets where Weighted-NCC achieves a significantly better performance than CS-NCC is much higher than the number of datasets in which Weighted-NCC is significantly outperformed by CS-NCC. Hence, for imprecise predictions, Weighted-NCC predicts fewer class states than CS-NCC.

To analyze the overall performance of CS-NCC and Weighted-NCC, Table 7 presents a summary of the MIC results. Similar to the tables associated with the other metrics, it allows us to see, for each cost matrix, the average MIC results obtained by CS-NCC and Weighted-NCC, in how many datasets a method obtains significantly better results than the other one via the Corrected Paired t-test, and whether an algorithm significantly outperforms the

Table 6: Summary of Indeterminacy Size results for each cost matrix. In the "Wilcoxon test" rows, "*" means that the method of the column significantly outperforms the other one through the Wilcoxon test for the corresponding cost matrix. The rows "Paired t-test" show the number of datasets in which the algorithm of the column performs significantly better than the other one via the Corrected Paired t-test for the corresponding cost matrix.

		CS-NCC	Weighted-NCC
Cost Matrix 0/1:	Wilcoxon test		*
	Paired t-test	0	**16**
	Average	7.8306	**5.621**
Cost Matrix (I):	Wilcoxon test		*
	Paired t-test	0	**27**
	Average	7.9163	**5.2825**
Cost Matrix (II):	Wilcoxon test		*
	Paired t-test	2	**19**
	Average	6.6141	**5.4406**
Cost Matrix (III):	Wilcoxon test		*
	Paired t-test	0	**16**
	Average	7.8591	**5.4478**
Cost Matrix (IV):	Wilcoxon test		*
	Paired t-test	0	**16**
	Average	6.2478	**5.3297**

other one by means of the Wilcoxon test. Again, we mark in bold the best results obtained in Average and Corrected Paired t-test. Figure 7 shows the average MIC results for each cost matrix. The complete MIC results can be found in Appendix A.

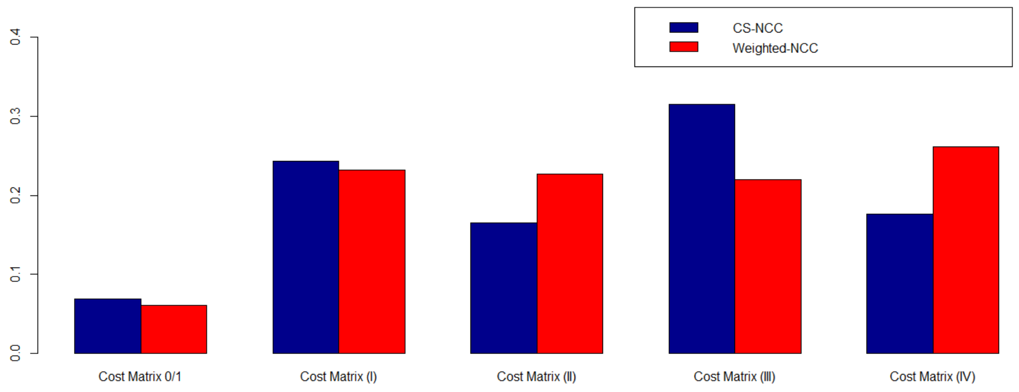

Figure 3: Average Single Cost results for each cost matrix.

Table 7: Summary of MIC results for each cost matrix. In the "Wilcoxon test" rows, "*" means that the method of the column significantly outperforms the other one through the Wilcoxon test for the corresponding cost matrix. The rows "Paired t-test" show the number of datasets in which the algorithm of the column performs significantly better than the other one via the Corrected Paired t-test for the corresponding cost matrix.

		CS-NCC	Weighted-NCC
Cost Matrix 0/1:	Wilcoxon test	=	=
	Paired t-test	10	**11**
	Average	0.8864	**0.9224**
Cost Matrix (I):	Wilcoxon test		*
	Paired t-test	3	**17**
	Average	0.8086	**0.9316**
Cost Matrix (II):	Wilcoxon test	=	=
	Paired t-test	**14**	10
	Average	0.8299	**0.8377**
Cost Matrix (III):	Wilcoxon test	=	=
	Paired t-test	**13**	10
	Average	**0.8461**	0.7471
Cost Matrix (IV):	Wilcoxon test		*
	Paired t-test	3	**20**
	Average	0.7679	**0.8733**

We remark the following issues related to these results:

- For Cost Matrices 0/1, (II), and (III), CS-NCC and Weighted-NCC

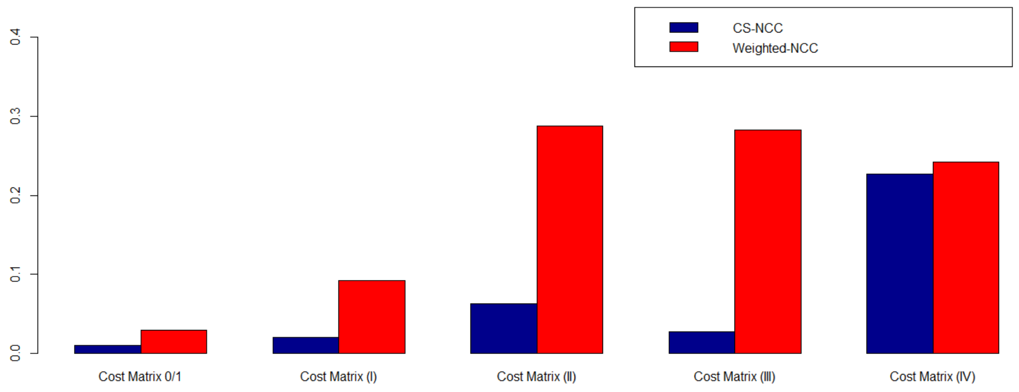

Figure 4: Average Set Cost results for each cost matrix.

obtain statistically equivalent results via the Wilcoxon test. According to the Corrected Paired t-test, for these cost matrices, the number of datasets in which CS-NCC significantly outperforms Weighted-NCC is not far greater than the number of datasets where the opposite occurs. For Cost Matrices 0/1 and (II), the average MIC value of Weighted-NCC is higher than the one achieved by CS-NCC. On the contrary, for Cost Matrix (III), CS-NCC gets a far greater average MIC value than Weighted-NCC. Thus, for Cost Matrices 0/1, (II), and (III), there is no clear winner between CS-NCC and Weighted-NCC.

- Weighted-NCC performs significantly better than CS-NCC via the Wilcoxon test for Cost Matrices (I) and (IV). Moreover, for these cost matrices, the number of datasets in which Weighted-NCC significantly outperforms CS-NCC via the Corrected Paired t-test is much larger than the number of datasets where CS-NCC performs significantly better than Weighted-NCC through this test. Also, for Cost Matrices (I) and (IV), the average MIC value achieved by Weighted-NCC is much higher than the one of CS-NCC (See Figure 7). So, for such cost matrices, Weighted-NCC performs far better than CS-NCC.

4.3. Processing time

In this subsection, we compare the processing time of CS-NCC, Weighted-NCC, and the original Naïve Bayes algorithm (NB). Remark that NB has

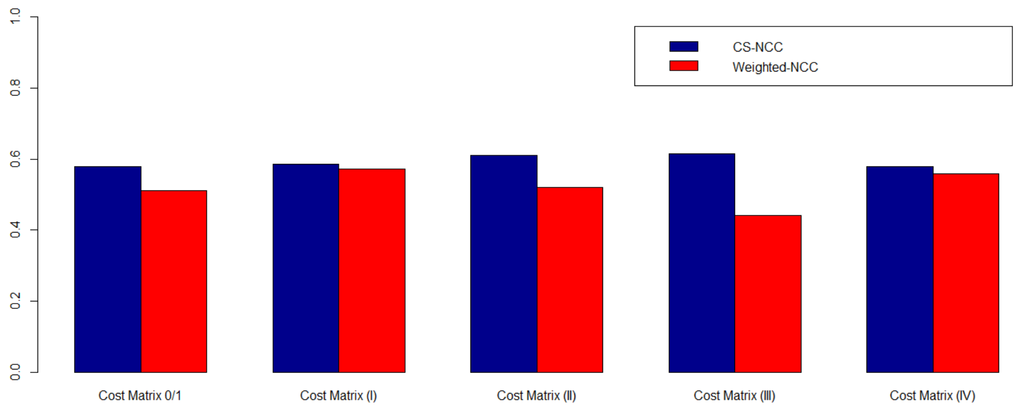

Figure 5: Average Determinacy results for each cost matrix.

been successfully applied to a massive amount of real data [59, 60, 61]. Thus, we aim to check the applicability of the proposed method to large databases. We cannot compare the performance of NB with the other two algorithms because, so far, there is no evaluation metric to compare the performance of a precise classifier with an imprecise classifier.

We utilize the same 34 datasets employed in the first part of the experiments using the same preprocessing. We have employed the implementation available in Weka for NB with default parameters. For both CS-NCC and Weighted-NCC, we have used Cost Matrix (I), assuming that the processing times of both algorithms with the other cost matrices are similar.

Following the recommendations given in [57] for statistical comparisons between three or more algorithms on many datasets, we compare the processing time of the three methods via the Friedman test [62] with a significance level of $\alpha = 0.05$. This test separately ranks the algorithms for each dataset. The null hypothesis of this test is that all algorithms perform equivalently.

Table 8 shows the average Friedman rank of each algorithm. The complete results related to processing time can be seen in Appendix B. According to the Friedman test, the three algorithms have equivalent performance with regard to processing time. In fact, we may observe that the average Friedman ranks of the three algorithms are quite similar. In consequence, we can state that NB, CS-NCC, and Weighted-NCC have similar processing times.

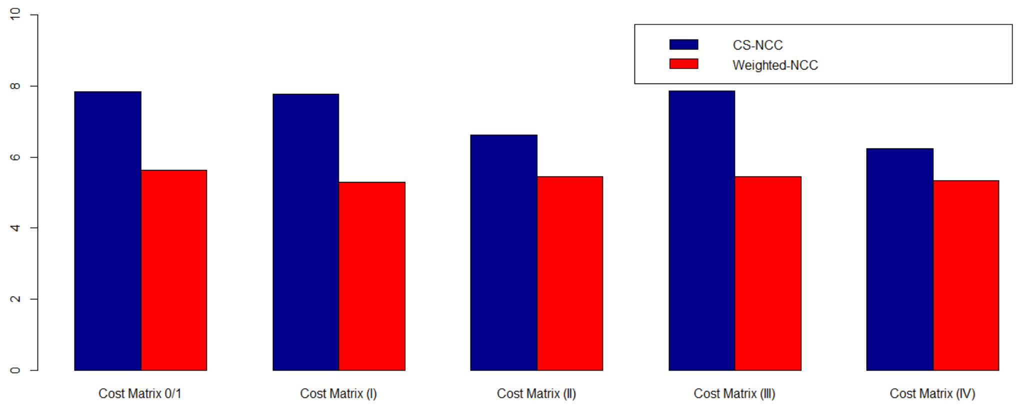

Figure 6: Average Indeterminacy Size results for each cost matrix.

Table 8: Average Friedman ranks of NB, CS-NCC, and Weighted-NCC corresponding to processing time.

Algorithm	Average Friedman rank
NB	2.0882
CS-NCC	1.9706.
Weighted-NCC	1.9402

4.4. Summary of the results

We summarize the experimental results as follows:

- CS-NCC predicts a single class state more frequently than Weighted-NCC. For instances precisely classified, the misclassification costs of Weighted-NCC tend to be lower than the misclassification costs of CS-NCC. It is because, as illustrated in Example 4, in some cases where it may be more intuitive to predict more than one class value, CS-NCC precisely classifies an instance, unlike Weighted-NCC.

- In contrast, for instances imprecisely classified, Weighted-NCC predicts fewer class states than CS-NCC even though the misclassification costs are higher with Weighted-NCC. This occurs since, as argued in Section 3.2 and shown in Example 3, Weighted-NCC might lead to more informative and intuitive predictions than CS-NCC.

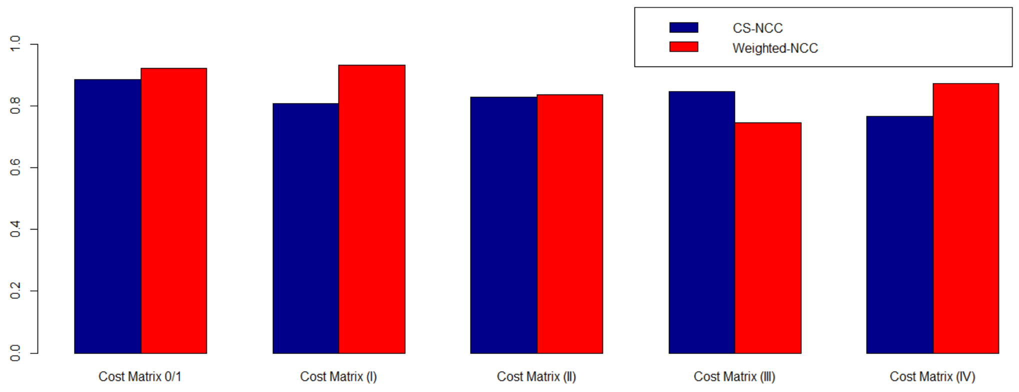

Figure 7: Average MIC results for each cost matrix.

- Concerning the whole performance, for Cost Matrices 0/1, (II), and (III), it can be stated that CS-NCC and Weighted-NCC obtain equivalent results. However, for Cost Matrices (I) and (IV), Weighted-NCC significantly outperforms CS-NCC. We may note that, for such cost matrices, predicting the more frequent class values may imply a higher cost than predicting the less frequent ones. In these cases, the risk intervals predicted by CS-NCC are probably less informative. Actually, for Cost Matrices (I) and (IV), CS-NCC does not significantly outperform Weighted-NCC through the Wilcoxon test in Determinacy. This might be the reason why, for these cost matrices, Weighted-NCC achieves a significantly better overall performance than CS-NCC.

- Therefore, we can conclude that, depending on the matrix of error costs considered, Weighted-NCC performs equivalently or significantly better than CS-NCC.

- Both CS-NCC and Weighted-NCC have equivalent processing time to NB. Consequently, our proposal can be applied to large databases, where the NB algorithm has been successfully employed.

5. Conclusions and future work

Bayesian classification methods are simple and effective and have been successfully utilized in practical applications. The main advantage of Bayesian classification algorithms is their low computational time. In consequence, they are quite suitable for large datasets, which is very important nowadays because of the huge amount of data in many areas. A quick and very known Bayesian classification method is the Naïve Bayes algorithm (NB). Despite its simplicity, NB has achieved good results in practice. In practical applications, misclassifications often imply different costs. Furthermore, for classifying an instance, classification algorithms usually predict a unique class value. Nevertheless, sometimes, due to the lack of information, it might more reasonable that classification methods predict two or more class values, which is known as Imprecise Classification (IC). In these situations, an imprecise classifier can provide different and vital information (see Example 1, which illustrates the importance of IC). Abellán and Masegosa [38] proposed a version of the NB algorithm for cost-sensitive IC, called cost-sensitive Naïve Credal Classifier (CS-NCC). So far, it is the only Bayesian method for cost-sensitive IC.

A Bayesian method for cost-sensitive IC, called the Weighted Naïve Credal Classifier (Weighted-NCC), has been introduced in this research. Weighted-NCC weights the instances by considering the error costs of their class values. For classifying an instance, Weighted-NCC estimates, for each class value, an interval score by using the Non-Parametric Predictive Inference Model, the instance weights, and the naïve assumption. Then, it determines the non-dominated states set by means of a dominance criterion on such intervals. We have shown that our proposed Weighted-NCC might yield more intuitive and informative predictions than the existing CS-NCC. Also, we have illustrated that, in some cases in which it may be more intuitive to predict more than one class value, CS-NCC predicts a single class state, unlike Weighted-NCC.

We have compared the performance of CS-NCC and our developed Weighted-NCC via an experimental analysis. Such an analysis has demonstrated that CS-NCC precisely classifies more instances than Weighted-NCC, but the misclassification costs for precise predictions are generally lower with the latter algorithm than with the former; for imprecise predictions, our proposed Weighted-NCC method predicts fewer class values than the existing CS-NCC even though the misclassification costs are higher with our proposal; regarding the whole performance, for some cost matrices, CS-NCC and Weighted-NCC obtain statistically equivalent results while, for other cost matrices, Weighted-NCC performs significantly better than CS-NCC.

Hence, we can conclude that the two Bayesian approaches for cost-sensitive IC are equally appropriate for some error cost settings. However, for other error cost settings, Weighted-NCC is much more suitable than CS-NCC. One of these settings is Cost Matrix (I), where the misclassification costs depend on the real values of the class variable, and less frequent class states have higher costs than more frequent values of the class variable. This type of error cost setting is employed in some essential fields, such as medicine. In this way, our proposed Bayesian algorithm for cost-sensitive IC is more suitable than the existing one for medical diagnosis to save lives.

Moreover, we have compared the processing time of CS-NCC, Weighted-NCC, and the original NB algorithm, which has been successfully applied to very large real datasets. It has been highlighted that the processing times of these three methods are equivalent. Consequently, our proposed algorithm may be appropriate for large databases. This is an important issue because of the huge amount of data in many application areas nowadays.

For future work, other Bayesian methods for cost-sensitive IC could be introduced by using the costs of incorrect classifications differently, proposing other estimations of the interval scores for the class values, or relaxing the naïve assumption. Our introduced method could also be adapted for other special types of classification. An example of such types is Multi-Label Classification, where some of the most recent algorithms can be found in [63, 64]. Moreover, it would be interesting to develop evaluation metrics to compare the performance of a precise classifier with an imprecise classifier.

Appendix A. Complete MIC results

Table 9: Complete MIC results for Cost Matrix (0/1). In the last column, ○ (●) means that Weighted-NCC (CS-NCC) significantly outperforms CS-NCC (Weighted-NCC) via the Corrected Paired t-test for the dataset of the row. The best result for each dataset is marked in bold font.

Dataset	CS-NCC	Weighted-NCC	
anneal	0.0054	**0.9655**	○
arrhythmia	0.0000	**0.4272**	○
audiology	0.0000	**0.2145**	○
autos	0.0000	**0.4342**	○
balance-scale	**0.6905**	0.6839	
bridges-version1	**0.2129**	0.0587	●
bridges-version2	**0.2285**	0.0594	●
car	1.1293	**1.1322**	
cmc	0.2830	**0.2861**	
dermatology	**1.5057**	1.3667	●
ecoli	**1.5936**	1.5800	
flags	0.0000	0.0000	
hypothyroid	**1.1946**	1.1386	●
iris	**0.9807**	0.9762	
letter	2.4024	**2.4356**	○
lymphography	**0.6639**	0.4502	●
mfeat-pixel	**1.4655**	0.8338	●
nursery	**1.4165**	1.2847	
optdigits	2.0918	**2.1136**	○
page-blocks	1.4751	**1.4856**	○
pendigits	1.9911	**2.0110**	○
postoperative-patient-data	**0.1408**	0.0286	●
primary-tumor	0.0004	**0.4545**	○
segment	1.7198	**1.7217**	
soybean	**2.5364**	2.4291	●
spectrometer	0.1088	**0.3590**	○
splice	**1.0217**	1.0215	
sponge	0.0000	0.0000	
tae	0.2589	**0.2608**	
vehicle	0.6697	**0.6782**	
vowel	1.1191	**1.4151**	○
waveform	0.7691	**0.7702**	
wine	**0.9854**	0.9142	●
zoo	**1.4764**	1.3721	●
Average	0.8864	**0.9224**	

Table 10: Complete MIC results for Cost Matrix (I). In the last column, ○ (●) means that Weighted-NCC (CS-NCC) significantly outperforms CS-NCC (Weighted-NCC) via the Corrected Paired t-test for the dataset of the row. The best result for each dataset is marked in bold font.

Dataset	CS-NCC	Weighted-NCC
anneal	0.0000	**1.0209** ○
arrhythmia	0.0000	**0.4629** ○
audiology	0.0000	**0.1677** ○
autos	0.0000	**0.5267** ○
balance-scale	**0.6037**	**0.6037**
bridges-version1	**0.1654**	0.0944
bridges-version2	**0.1331**	0.1061
car	**1.0843**	1.0748 ●
cmc	0.1062	**0.1094**
dermatology	**1.4902**	1.4471
ecoli	1.4362	**1.5615** ○
flags	0.0000	0.0000
hypothyroid	1.1189	**1.3431** ○
iris	0.9205	**0.9208**
letter	2.0222	**2.0411** ○
lymphography	0.5647	**0.8745** ○
mfeat-pixel	**1.4528**	1.2027 ●
nursery	**1.3718**	1.3613 ●
optdigits	2.0249	**2.0555** ○
page-blocks	1.4512	**1.4699** ○
pendigits	1.8589	**1.8760** ○
postoperative-patient-data	**0.0108**	0.0003
primary-tumor	0.0000	**0.5915** ○
segment	1.6812	**1.6942**
soybean	**2.4724**	2.4480
spectrometer	0.0993	**0.7478** ○
splice	1.0036	**1.0082** ○
sponge	0.0000	**0.1097** ○
tae	-0.0481	**-0.0406**
vehicle	0.4522	**0.4942** ○
vowel	0.8738	**1.0713** ○
waveform	0.7084	**0.7097**
wine	**0.9835**	0.9671
zoo	1.4493	**1.5528**
Average	0.8086	**0.9316**

Table 11: Complete MIC results for Cost Matrix (II). In the last column, ○ (●) means that Weighted-NCC (CS-NCC) significantly outperforms CS-NCC (Weighted-NCC) via the Corrected Paired t-test for the dataset of the row. The best result for each dataset is marked in bold font.

Dataset	CS-NCC	Weighted-NCC
anneal	0.0179	**0.9405** ○
arrhythmia	0.0897	**0.2725** ○
audiology	0.1627	**0.2190**
autos	0.0000	**0.2343** ○
balance-scale	0.6609	**0.6702**
bridges-version1	**0.2715**	0.1268 ●
bridges-version2	**0.2740**	0.1046 ●
car	0.8639	**1.0668** ○
cmc	**0.1340**	0.0299 ●
dermatology	**1.4816**	1.4109 ●
ecoli	1.4671	**1.5120**
flags	0.0000	0.0000
hypothyroid	**1.2501**	1.0742 ●
iris	0.9328	**0.9457**
letter	1.7340	**2.0425** ○
lymphography	**0.7337**	0.2505 ●
mfeat-pixel	**1.4569**	1.2063 ●
nursery	**1.2319**	1.1556
optdigits	2.0142	**2.0476** ○
page-blocks	**1.4665**	1.4533 ●
pendigits	1.8371	**1.8803** ○
postoperative-patient-data	**0.4163**	-0.1210 ●
primary-tumor	**0.3412**	0.0732 ●
segment	1.6444	**1.6809** ○
soybean	**2.5135**	2.4341 ●
spectrometer	0.1204	**0.2521**
splice	0.9977	**1.0040** ○
sponge	0.0000	0.0000
tae	**0.0151**	0.0114
vehicle	**0.4371**	0.4152
vowel	0.5418	**1.1511** ○
waveform	**0.6549**	0.6379 ●
'wine	**0.9849**	0.9325 ●
zoo	**1.4702**	1.3680 ●
Average	0.8299	**0.8377**

Table 12: Complete MIC results for Cost Matrix (III). In the last column, ○ (●) means that Weighted-NCC (CS-NCC) significantly outperforms CS-NCC (Weighted-NCC) via the Corrected Paired t-test for the dataset of the row. The best result for each dataset is marked in bold font.

Dataset	CS-NCC	Weighted-NCC
anneal	0.0679	**0.7941** ○
arrhythmia	0.1362	**0.2371** ○
audiology	**0.2157**	0.1729
autos	0.0000	**0.2078** ○
balance-scale	**0.5742**	0.5739
bridges-version1	**0.2480**	0.1261 ●
bridges-version2	**0.2666**	0.1117 ●
car	0.9102	**0.9232** ○
cmc	0.0278	**0.0449** ○
dermatology	**1.5080**	1.2305 ●
ecoli	**1.4616**	1.4257
flags	0.0000	0.0000
hypothyroid	**1.2713**	1.0505 ●
iris	**0.9809**	0.9787
letter	2.0300	**2.0581** ○
lymphography	**0.7196**	0.1348 ●
mfeat-pixel	**1.4600**	1.0841 ●
nursery	**1.3058**	0.6901 ●
optdigits	2.0097	**2.0230** ○
page-blocks	**1.4184**	1.3962 ●
pendigits	1.8768	**1.9063** ○
postoperative-patient-data	**0.3024**	-0.0336 ●
primary-tumor	**0.2514**	0.0561 ●
segment	1.6630	**1.6770**
soybean	**2.4414**	1.3424 ●
spectrometer	**0.1098**	0.1012
splice	0.9970	**0.9974**
sponge	0.0000	0.0000
tae	**0.1043**	0.0993
vehicle	0.4226	**0.4337**
vowel	0.8782	**1.1050** ○
waveform	0.6312	**0.6364** ○
wine	**0.9860**	0.9141 ●
zoo	**1.4918**	0.9029 ●
Average	**0.8461**	0.7471

Table 13: Complete MIC results for Cost Matrix (IV). In the last column, ○ (●) means that Weighted-NCC (CS-NCC) significantly outperforms CS-NCC (Weighted-NCC) via the Corrected Paired t-test for the dataset of the row. The best result for each dataset is marked in bold font.

Dataset	CS-NCC	Weighted-NCC
anneal	0.8050	**0.9717** ○
arrhythmia	0.0398	**0.2857** ○
audiology	0.0885	**0.2210** ○
autos	0.0229	**0.3561** ○
balance-scale	0.5077	**0.5578**
bridges-version1	**0.1803**	0.0407 ●
bridges-version2	**0.1715**	0.0595 ●
car	0.6703	**1.0635** ○
cmc	0.0425	**0.1026** ○
dermatology	**1.4868**	1.4662
ecoli	1.1761	**1.4621** ○
flags	0.0000	0.0000
hypothyroid	1.2193	**1.2226**
iris	**0.9643**	0.9634
letter	1.7413	**2.0408** ○
lymphography	0.5142	**0.5863** ○
mfeat-pixel	**1.4549**	1.3141 ●
nursery	1.0020	**1.2771** ○
optdigits	1.9908	**2.0376** ○
page-blocks	1.4033	**1.4358** ○
pendigits	1.8610	**1.8974** ○
postoperative-patient-data	-0.1263	**-0.0979**
primary-tumor	-0.2258	**0.1229** ○
segment	**1.7046**	1.7036
soybean	2.3312	**2.4594** ○
spectrometer	0.0617	**0.2815** ○
splice	0.9985	**1.0050**
sponge	0.0000	**0.0241** ○
tae	0.0096	**0.0705**
vehicle	0.3803	**0.4905** ○
vowel	0.5198	**1.1512** ○
waveform	0.6809	**0.6986** ○
wine	**0.9834**	0.9725
zoo	1.4476	**1.4485**
Average	0.7679	**0.8733**

Appendix B. Complete processing time results

Table 14: Complete processing time results for NB, CS-NCC, and Weighted-NCC, in seconds. The best result for each dataset is marked in bold font.

Dataset	NB	CS-NCC	Weighted-NCC
anneal	0.0070	**0.0047**	00.0048
arrhythmia	0.0431	0.0378	**0.0361**
audiology	0.0014	0.0013	**0.0008**
autos	0.0027	**0.0022**	0.0025
balance-scale	0.0009	0.0011	**0.0008**
bridges-version1	0.0006	**0.0003**	0.0005
bridges-version2	0.0002	**0.0000**	0.0003
car	**0.0008**	0.0011	0.0013
cmc	**0.0009**	0.0013	**0.0009**
dermatology	0.0011	**0.0005**	0.0008
ecoli	**0.0006**	0.0009	0.0013
flags	0.0005	0.0005	**0.0002**
hypothyroid	0.0148	0.0119	**0.0105**
iris	**0.0003**	0.0005	**0.0003**
letter	0.2061	0.2030	**0.2014**
lymphography	**0.0003**	**0.0003**	0.0005
mfeat-pixel	**0.0131**	0.0133	0.0133
nursery	0.0098	0.0123	**0.0088**
optdigits	0.1100	**0.0986**	0.1202
page-blocks	**0.0394**	0.0438	0.0408
pendigits	0.0969	**0.0919**	0.1061
postoperative-patient-data	**0.0002**	0.0003	**0.0002**
primary-tumor	**0.0003**	0.0006	0.0005
segment	**0.0355**	0.0378	0.0359
soybean	0.0008	0.0008	0.0008
spectrometer	0.2906	**0.2633**	0.2658
splice	0.0064	**0.0047**	0.0053
sponge	**0.0003**	0.0005	0.0005
tae	**0.0000**	**0.0000**	0.0005
vehicle	0.0052	**0.0045**	0.0047
vowel	0.0172	0.0178	**0.0169**
waveform	0.0745	**0.0728**	0.0794
wine	0.0016	0.0017	**0.0014**
zoo	0.0002	**0.0000**	**0.0000**
Average	0.0289	**0.0274**	0.0283

References

[1] A. Behler, H.-P. Müller, A. C. Ludolph, D. Lulé, J. Kassubek, A multivariate bayesian classification algorithm for cerebral stage prediction by diffusion tensor imaging in amyotrophic lateral sclerosis, NeuroImage: Clinical 35 (2022) 103094. doi:10.1016/j.nicl.2022.103094.

[2] H. Chen, S. Hu, R. Hua, X. Zhao, Improved naive bayes classification algorithm for traffic risk management, EURASIP Journal on Advances in Signal Processing 2021 (1) (2021) 30. doi:10.1186/s13634-021-00742-6.

[3] D. van Herwerden, J. W. O'Brien, P. M. Choi, K. V. Thomas, P. J. Schoenmakers, S. Samanipour, Naive bayes classification model for isotopologue detection in lc-hrms data, Chemometrics and Intelligent Laboratory Systems 223 (2022) 104515. doi:10.1016/j.chemolab.2022.104515.

[4] S. Y. Yerima, S. Sezer, G. McWilliams, I. Muttik, A new android malware detection approach using bayesian classification, in: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 2013, pp. 121–128. doi:10.1109/AINA.2013.88.

[5] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, New York, 1973.

[6] I. Wickramasinghe, H. Kalutarage, Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation., Soft Computing 25 (2020) 2277–2293. doi:10.1007/s00500-020-05297-6.

[7] T. M. Ma, K. Yamamori, A. Thida, A comparative approach to naïve bayes classifier and support vector machine for email spam classification, in: 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), 2020, pp. 324–326. doi:10.1109/GCCE50665.2020.9291921.

[8] R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, M. R. Sillero-Denamiel, Variable selection for naïve bayes classification, Computers & Operations Research 135 (2021) 105456. doi:10.1016/j.cor.2021.105456.

[9] D.-H. Vu, Privacy-preserving naive bayes classification in semi-fully distributed data model, Computers & Security 115 (2022) 102630. doi:10.1016/j.cose.2022.102630.

[10] S. Bakheet, A. Al-Hamadi, A framework for instantaneous driver drowsiness detection based on improved hog features and naïve bayesian classification, Brain Sciences 11 (2) (2021) 240. doi:10.3390/brainsci11020240.

[11] V. Jackins, S. Vimal, M. Kaliappan, M. Y. Lee, Ai-based smart prediction of clinical disease using random forest classifier and naive bayes, The Journal of Supercomputing 77 (5) (2021) 5198–5219. doi:10.1007/s11227-020-03481-x.

[12] L. M. de Campos, A. Cano, J. G. Castellano, S. Moral, Bayesian networks classifiers for gene-expression data, in: 11th International Conference on Intelligent Systems Design and Applications, 2011, pp. 1200–1206. doi:10.1109/ISDA.2011.6121822.

[13] M. Zaffalon, The naive credal classifier, Journal of Statistical Planning and Inference 105 (1) (2002) 5 – 21. doi:10.1016/S0378-3758(01)00201-4.

[14] G. Corani, M. Zaffalon, Learning reliable classifiers from small or incomplete data sets: the naive credal classifier 2, Journal of Machine Learing Research 9 (2008) 581–621.

[15] P. Walley, Inferences from multinomial data; learning about a bag of marbles (with discussion), Journal of the Royal Statistical Society. Series B (Methodological) 58 (1) (1996) 3–57. doi:10.2307/2346164.

[16] B. Zhao, M. Yang, H. Diao, B. An, Y. Zhao, Y. Zhang, A novel approach to transformer fault diagnosis using idm and naive credal classifier, International Journal of Electrical Power & Energy Systems 105 (2019) 846 – 855. doi:10.1016/j.ijepes.2018.09.029.

[17] A. Antonucci, G. Corani, The multilabel naive credal classifier, International Journal of Approximate Reasoning 83 (2017) 320–336. doi:10.1016/j.ijar.2016.10.006.

[18] R. Bhuvaneswari, K. Kalaiselvi, Naive bayesian classification approach in healthcare applications, International Journal of Computer Science and Telecommunications 3 (1) (2012) 106–112.

[19] D. Zhu, H. Zhu, X. Liu, H. Li, F. Wang, H. Li, D. Feng, Credo: Efficient and privacy-preserving multi-level medical pre-diagnosis based on ml-knn, Information Sciences 514 (2020) 244–262. doi:10.1016/j.ins.2019.11.041.

[20] D. Gan, J. Shen, B. An, M. Xu, N. Liu, Integrating tanbn with cost sensitive classification algorithm for imbalanced data in medical diagnosis, Computers & Industrial Engineering 140 (2020) 106266. doi:10.1016/j.cie.2019.106266.

[21] I. D. Mienye, Y. Sun, Performance analysis of cost-sensitive learning methods with application to imbalanced medical data, Informatics in Medicine Unlocked 25 (2021) 100690. doi:10.1016/j.imu.2021.100690.

[22] G. Shen, Z. Fu, Y. Gui, W. Susilo, M. Zhang, Efficient and privacy-preserving online diagnosis scheme based on federated learning in e-healthcare system, Information Sciences 647 (2023) 119261. doi:10.1016/j.ins.2023.119261.

[23] S. Akila, U. Srinivasulu Reddy, Cost-sensitive risk induced bayesian inference bagging (ribib) for credit card fraud detection, Journal of Computational Science 27 (2018) 247–254. doi:10.1016/j.jocs.2018.06.009.

[24] S. Nami, M. Shajari, Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors, Expert Systems with Applications 110 (2018) 381–392. doi:10.1016/j.eswa.2018.06.011.

[25] J. Forough, S. Momtazi, Ensemble of deep sequential models for credit card fraud detection, Applied Soft Computing 99 (2021) 106883. doi:10.1016/j.asoc.2020.106883.

[26] O. F. Arar, K. Ayan, Software defect prediction using cost-sensitive neural network, Applied Soft Computing 33 (2015) 263–277. doi:10.1016/j.asoc.2015.04.045.

[27] T. Zivkovic, B. Nikolic, V. Simic, D. Pamucar, N. Bacanin, Software defects prediction by metaheuristics tuned extreme gradient boosting and analysis based on shapley additive explanations, Applied Soft Computing 146 (2023) 110659. doi:10.1016/j.asoc.2023.110659.

[28] L. Niu, J. Wan, H. Wang, K. Zhou, Cost-sensitive dictionary learning for software defect prediction, Neural Processing Letters 52 (3) (2020) 2415–2449. doi:10.1007/s11063-020-10355-z.

[29] F. Jiang, X. Yu, D. Gong, J. Du, A random approximate reduct-based ensemble learning approach and its application in software defect prediction, Information Sciences 609 (2022) 1147–1168. doi:https://doi.org/10.1016/j.ins.2022.07.130.

[30] S. K. Biswas, M. Chakraborty, H. R. Singh, D. Devi, B. Purkayastha, A. K. Das, Hybrid case-based reasoning system by cost-sensitive neural network for classification, Soft Computing 21 (24) (2017) 7579–7596.

[31] H. Li, L. Zhang, X. Zhou, B. Huang, Cost-sensitive sequential three-way decision modeling using a deep neural network, International Journal of Approximate Reasoning 85 (2017) 68–78. doi:10.1016/j.ijar.2017.03.008.

[32] K. M. Ting, An instance-weighting method to induce cost-sensitive trees, IEEE Transactions on Knowledge and Data Engineering 14 (3) (2002) 659–665. doi:10.1109/TKDE.2002.1000348.

[33] C. Qiu, L. Jiang, C. Li, Randomly selected decision tree for test-cost sensitive learning, Applied Soft Computing 53 (2017) 27–33. doi:0.1016/j.asoc.2016.12.047.

[34] Z. Li, J. Zhang, X. Yao, G. Kou, How to identify early defaults in online lending: A cost-sensitive multi-layer learning framework, Knowledge-Based Systems 221 (2021) 106963. doi:10.1016/j.knosys.2021.106963.

[35] G. M. Di Nunzio, A new decision to take for cost-sensitive naive bayes classifiers, Information Processing & Management 50 (5) (2014) 653–674. doi:10.1016/j.ipm.2014.04.008.

[36] Y. Alsubaie, K. E. Hindi, H. Alsalman, Cost-sensitive prediction of stock price direction: Selection of technical indicators, IEEE Access 7 (2019) 146876–146892. doi:10.1109/ACCESS.2019.2945907.

[37] Y. Xiong, M. Ye, C. Wu, Cancer classification with a cost-sensitive naive bayes stacking ensemble, Computational and Mathematical Methods in Medicine 1 (1) (2021) 5556992. doi:10.1155/2021/5556992.

[38] J. Abellán, A. R. Masegosa, Imprecise classification with credal decision trees, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20 (05) (2012) 763–787. doi:10.1142/S0218488512500353.

[39] J. Abellán, C. J. Mantas, J. G. Castellano, AdaptativeCC4.5: Credal C4.5 with a rough class noise estimator, Expert Systems with Applications 92 (Supplement C) (2018) 363 – 379. doi:10.1016/j.eswa.2017.09.057.

[40] F. P. A. Coolen, T. Augustin, Learning from multinomial data: a nonparametric predictive alternative to the imprecise dirichlet model, in: ISIPTA'05: Proceedings of the Fourth International Symposium on Imprecise Probabilities and their Applications, Fabio G. Cozman, Robert Nau and Teddy Seidenfeld (Editors)., 2005, pp. 125–134.

[41] F. Coolen, T. Augustin, A nonparametric predictive alternative to the imprecise dirichlet model: The case of a known number of categories, International Journal of Approximate Reasoning 50 (2) (2009) 217 – 230. doi:10.1016/j.ijar.2008.03.011.

[42] J. Abellán, R. M. Baker, F. P. Coolen, R. J. Crossman, A. R. Masegosa, Classification with decision trees from a nonparametric predictive inference perspective, Computational Statistics & Data Analysis 71 (2014) 789 – 802. doi:10.1016/j.csda.2013.02.009.

[43] S. Moral, C. J. Mantas, J. G. Castellano, J. Abellán, Imprecise classification with non-parametric predictive inference, in: M.-J. Lesot, S. Vieira, M. Z. Reformat, J. P. Carvalho, A. Wilbik, B. Bouchon-Meunier, R. R. Yager (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer International Publishing, 2020, pp. 53–66.

[44] J. Abellán, Uncertainty measures on probability intervals from the imprecise dirichlet model, International Journal of General Systems 35 (5) (2006) 509–528. doi:10.1080/03081070600687643.

[45] J. Abellán, R. M. Baker, F. P. Coolen, Maximising entropy on the nonparametric predictive inference model for multinomial data, European Journal of Operational Research 212 (1) (2011) 112 – 122. doi:10.1016/j.ejor.2011.01.020.

[46] J. Abellán, Equivalence relations among dominance concepts on probability intervals and general credal sets, International Journal of General Systems 41 (2) (2012) 109–122. doi:10.1080/03081079.2011.607449.

[47] W. Y. Loh, Classification and regression trees, WIREs Data Mining and Knowledge Discovery 1 (1) (2011) 14–23. doi:doi.org/10.1002/widm.8.

[48] G. J. Klir, Uncertainty and Information: Foundations of Generalized Information Theory, John Wiley And Sons, Inc., 2006. doi:10.1002/0471755575.

[49] S. Moral-García, J. Abellán, Uncertainty-based information measures on the approximate non-parametric predictive inference model, International Journal of General Systems 50 (2) (2021) 159–181. doi:10.1080/03081079.2020.1866567.

[50] C. Fleizach, A naive bayes classifier on 1998 kdd cup, 2006.
URL https://api.semanticscholar.org/CorpusID:12802005

[51] M. Lichman, UCI machine learning repository (2013).
URL http://archive.ics.uci.edu/ml

[52] S. Moral-García, J. Abellán, T. Coolen-Maturi, F. P. Coolen, A cost-sensitive imprecise credal decision tree based on nonparametric predictive inference, Applied Soft Computing 123 (2022) 108916. doi:10.1016/j.asoc.2022.108916.

[53] S. Moral-García, C. J. Mantas, J. G. Castellano, M. D. Benítez, J. Abellán, Bagging of credal decision trees for imprecise classification, Expert Systems with Applications 141 (2020) 112944. doi:10.1016/j.eswa.2019.112944.

[54] S. Moral-García, J. G. Castellano, C. J. Mantas, J. Abellán, Using extreme prior probabilities on the naive credal classifier, Knowledge-Based Systems 237 (2022) 107707. doi:10.1016/j.knosys.2021.107707.

[55] U. Fayyad, K. Irani, Multi-valued interval discretization of continuous-valued attributes for classification learning, in: Proceeding of the 13th International joint Conference on Artificial Inteligence, Morgan Kaufmann, 1993, pp. 1022–1027.

[56] I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[57] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[58] F. Wilcoxon, Individual Comparisons by Ranking Methods, Biometrics Bulletin 1 (6) (1945) 80–83. doi:10.2307/3001968.

[59] C. Banchhor, N. Srinivasu, Integrating cuckoo search-grey wolf optimization and correlative naive bayes classifier with map reduce model for big data classification, Data & Knowledge Engineering 127 (2020) 101788. doi:10.1016/j.datak.2019.101788.

[60] H. Chen, S. Hu, R. Hua, X. Zhao, Improved naive bayes classification algorithm for traffic risk management, EURASIP Journal on Advances in Signal Processing 2021 (1) (2021) 30. doi:10.1186/s13634-021-00742-6.

[61] S. K. Punia, M. Kumar, T. Stephan, G. G. Deverajan, R. Patan, Performance analysis of machine learning algorithms for big data classification: Ml and ai-based algorithms for big data analysis, International Journal of E-Health and Medical Communications (IJEHMC) 12 (4) (2021) 60–75. doi:10.4018/IJEHMC.20210701.oa4.

[62] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, The Annals of Mathematical Statistics 11 (1) (1940) 86–92. doi:10.1214/aoms/1177731944.

[63] T. Yin, H. Chen, J. Wan, P. Zhang, S.-J. Horng, T. Li, Exploiting feature multi-correlations for multilabel feature selection in robust multi-neighborhood fuzzy β covering space, Information Fusion 104 (2024) 102150. doi:0.1016/j.inffus.2023.102150.

[64] T. Yin, H. Chen, Z. Yuan, J. Wan, K. Liu, S.-J. Horng, T. Li, A robust multilabel feature selection approach based on graph structure considering fuzzy dependency and feature interaction, IEEE Transactions on Fuzzy Systems 31 (12) (2023) 4516–4528. doi:10.1109/TFUZZ.2023.3287193.