Direct Nonparametric Predictive Inference Classification Trees

Abdulmajeed Atiah Alharbi[a,b,*], Frank P.A. Coolen[b], Tahani Coolen-Maturi[b]

[a]*Department of Mathematics, Taibah University, Madinah, Saudi Arabia*
[b]*Department of Mathematical Sciences, Durham University, Durham, UK*

Abstract

Classification is the task of assigning a new instance to one of a set of predefined categories based on the attributes of the instance. A classification tree is one of the most commonly used techniques in the area of classification. In this paper, we introduce a novel classification tree algorithm which we call Direct Nonparametric Predictive Inference (D-NPI) classification algorithm. The D-NPI algorithm is completely based on the Nonparametric Predictive Inference (NPI) approach, and it does not use any other assumptions. NPI is a statistical methodology which learns from data in the absence of prior knowledge and uses only few modelling assumptions, enabled by the use of lower and upper probabilities to quantify uncertainty. Due to the predictive nature of NPI, it is well suited for classification, as the nature of classification is explicitly predictive as well. The D-NPI algorithm uses a new split criterion called Correct Indication (CI). CI reflects how informative attribute variables are, hence if the attribute variable is very informative, it gives high NPI lower and upper probabilities for CI. In addition, CI reports the strength of the evidence that the attribute variables will indicate regarding the possible class state for future instances, based on the data. To demonstrate its real-world applicability, the D-NPI algorithm is tested on benchmark data sets from various domains obtained from the UCI machine learning repository. The performance of the D-NPI classification algorithm is tested against several other classification algorithms using classification accuracy, in-sample accuracy and tree size. The experimental results indicate that the D-NPI classification algorithm performs well and tends to slightly outperform the other classification algorithms.

*Corresponding author
Email addresses: aahharbi@taibahu.edu.sa (Abdulmajeed Atiah Alharbi), frank.coolen@durham.ac.uk (Frank P.A. Coolen), tahani.maturi@durham.ac.uk (Tahani Coolen-Maturi)

Keywords: Nonparametric predictive inference, Imprecise probability, Correct indication, Classification, Classification trees

1. Introduction

Classification is one of the most common data mining techniques that is used for assigning a new instance to one of a set of predefined categories based on the attributes of the instance. The aim of classification is to predict the unknown class states of instances given known attribute values. There are many classification methods available in the literature, the classification tree is one of the most commonly used because of its interpretational simplicity. There are a number of algorithms that can be used to build classification trees. For example, the ID3 algorithm [38], the C4.5 algorithm [39] and another algorithm from an imprecise probability perspective [5].

In recent years, theories of imprecise probabilities have been widely developed for several areas of statistics. Many methods of statistical inference have been introduced based on imprecise probability theory, and it has been shown that they have some advantages over other methods based on the classical probability theory. Augustin et al. [9] have presented an overview of the main aspects of imprecise probability and its applications. Walley [42] has introduced the Imprecise Dirichlet Model (IDM) for inference based on multinomial data. The IDM has been used in several statistical problems [11], including classification [5]. However, the use of the IDM has been criticised for some drawbacks [37]. There have been several criticisms of the IDM [22]. One important issue is that the IDM assigns a lower probability of $1/(1+s)$ for the event where the second observation matches the first. Even when using small values of the parameter s in the IDM, like 1 or 2, this leads to a surprisingly high value for the lower probability. Additionally, the IDM predictive lower and upper probabilities are based solely on the observed frequency of that category and the total number of observations. Another limitation is its inability to differentiate between defined new categories and unobserved new outcomes when considering events that include unseen categories. Furthermore, the IDM's lower and upper probabilities for the event that the next observation falls into an unseen category do not rely on the number of categories observed so far [22]. An alternative approach for inference from multinomial data has been presented by Coolen and Augustin [22], which is Nonparametric Predictive Inference for Multinomial data (NPI-M).

Nonparametric Predictive Inference (NPI) is a frequentist statistical method which uses only few model assumptions, enabled by the use of lower and upper probabilities to quantify uncertainty. NPI has been developed in recent years for different applications in statistics, operations research, risk and reliability [16, 18, 21]. NPI has been introduced for several types of datasets, such as Binary data [19], real-valued data [8, 31], right-censored data [15, 24], ordinal data [27] and multinomial data [10, 22, 23]. NPI is based on Hill's assumption $A_{(n)}$ [28], which is used for prediction about future instances with real-valued data. The $A_{(n)}$ assumption is not suitable for multinomial data, hence, a variation of Hill's assumption $A_{(n)}$, which is called *circular-*$A_{(n)}$ assumption, is used for multinomial data [20].

Due to the predictive nature of NPI, it is well suited for classification, as the nature of classification is explicitly predictive as well. Therefore, several classification methods have been successfully developed based on the NPI approach [3, 10, 33, 34, 35]. Different classification trees have been built based on NPI using an extension of the information gain split criterion, which is a well-known classic split criterion used by the ID3 algorithm. These classification trees are built by replacing precise probabilities in the classical method with imprecise probabilities, which are obtained using the NPI approach. In this paper, we build classification trees completely based on NPI and without adding any further assumptions. This is achieved by introducing a new split criterion, which is based on the NPI lower and upper probabilities and does not use any other added concepts from the literature.

In this paper, we propose a new algorithm to build classification trees using imprecise probabilities and based on the NPI approach, which we call Direct Nonparametric Predictive Inference (D-NPI) classification tree algorithm. As a first step, we introduce the D-NPI classification algorithm for binary data. Thereafter, the D-NPI classification algorithm for multinomial data is presented. The D-NPI classification algorithm can base classification on the lower and upper probabilities for events with binary or multinomial data, without adding any further assumptions.

The rest of this paper is organized as follows: Section 2 briefly provides a background on classification trees with classic or imprecise split criteria and a brief overview of imprecise probability and the NPI approach. Section 3 introduces the direct classification approach using NPI for binary and multinomial data. Section 4 presents our new split criterion, Correct Indication, for both

binary and multinomial data. In Section 5, the proposed D-NPI classification algorithm is explained. Section 6 describes the experimental analysis carried out on different datasets to assess the proposed algorithm and compare it with other classification methods. Finally, conclusions and topics for future research are briefly discussed in Section 7.

2. Background

In this section, we review some of the most commonly used classic and imprecise split criteria that are used to build classification trees. Then, an introduction to imprecise probability is given. Finally, the Nonparametric Predictive Inference (NPI) method is introduced, particularly for binary and multinomial data.

2.1. Classification trees

A classification tree is a nonparametric technique which represents a hierarchical data structure. The use of a classification tree is to classify a new instance into one of a predefined set of classes based on its attributes' values. Classification trees are mainly used on a data set that contains one or more attribute variables and a categorical target variable. In a classification tree, each non-leaf node represents an attribute variable, each branch denotes the outcome of an attribute variable and each leaf node is assigned to one class label. Classifying a new instance is straightforward once a classification tree has been built. Instances are classified by navigating them from the root node of the tree and going down to a leaf node according to the value of the attribute variables along the path [30]. The leaf nodes correspond to classes that instances are assigned to them.

2.1.1. Split criteria

A classification tree algorithm requires a split criterion which is used to select the best attribute variable to split on at each step of building the tree. Several classification tree algorithms have been developed using different split criteria. Split criteria are mainly used in order to reduce the impurity of a node. We briefly review two of the most commonly used classic split criteria, which are Information Gain [38] and Information Gain Ratio [39]. These split criteria are used to implement the ID3 and the C4.5 algorithms, respectively. In addition, we briefly review an imprecise split criterion, called Imprecise Information Gain, that has been used to construct credal classification trees [5].

Information Gain: The *Information Gain* split criterion was introduced by Quinlan in 1986 [38] as a split criterion for the ID3 algorithm. Information Gain uses entropy as an impurity measure. Entropy, also called the Shannon Entropy [40], of a training set $\mathcal{D}$ is given by

$$H(\mathcal{D}) = -\sum_{i=1}^{m} p_i \log_2(p_i), \tag{1}$$

where p_i is the proportion of $\mathcal{D}$ belonging to class i (for $i = 1, ..., m$), so m is the total number of classes, and $\log_2$ is used because the information is coded in bits [17]. Generally speaking, entropy represents a level of uncertainty or impurity in a set of instances. The Information Gain of an attribute A, relative to the training set $\mathcal{D}$ is given by

$$Gain(\mathcal{D}, A) = H(\mathcal{D}) - \sum_{j=1}^{n} \frac{|\mathcal{D}_j|}{|\mathcal{D}|} H(\mathcal{D}_j), \tag{2}$$

where the training set $\mathcal{D}$ is partitioned into n partitions corresponding to the value of the attribute variable A, and $\mathcal{D}_j$ is the subset of $\mathcal{D}$ for which attribute A has value j, where $|\mathcal{D}|$ denotes the cardinality of the set $\mathcal{D}$. The Information Gain handles only categorical attributes.

Gain Ratio: The *Gain Ratio* split criterion was introduced by Quinlan in 1993 [39] as an extension to the Information Gain split criterion. It is used as a split criterion for the C4.5 algorithm. Unlike the ID3, the C4.5 algorithm handles both categorical and continuous attributes. The Information Gain is biased toward attribute variables that have many states [38]. So, these attribute variables are more likely to be selected. To solve this problem, Quinlan [39] introduced the Gain Ratio split criterion, which normalizes the Information Gain as follows:

$$GR(\mathcal{D}, A) = \frac{Gain(\mathcal{D}, A)}{SI(\mathcal{D}, A)}\ , \tag{3}$$

where $Gain(\mathcal{D}, A)$ is given by Equation (2), and Split Information $SI(\mathcal{D}, A)$ is given by:

$$SI(\mathcal{D}, A) = -\sum_{j=1}^{n} \frac{|\mathcal{D}_j|}{|\mathcal{D}|} \log_2 \frac{|\mathcal{D}_j|}{|\mathcal{D}|}. \tag{4}$$

The $SI(\mathcal{D}, A)$ represents the information generated by splitting the training data set $\mathcal{D}$ into n partitions corresponding to the values of the attribute vari-

able A. The C4.5 algorithm builds classification trees in a similar way to the ID3 algorithm, but it uses the Gain Ratio split criterion (Formula 3) to select the splitting attribute variable at each node.

Imprecise Information Gain: The *Imprecise Information Gain* (IIG) split criterion was introduced by Abellán and Moral in 2003 [5] to build classification trees from an imprecise probability perspective. The IIG for an attribute variable A is defined as follows:

$$IIG(A, C) = S(K(C)) - \sum_i p(a_i) S(K(C|(A = a_i))), \tag{5}$$

where $S(K)$ is the maximum entropy of a credal set, and $K(C)$ and $K(C|(A = a_i))$ are credal sets for the class variable C and for C given the value a_i of the attribute variable A, respectively; and $i = 1, ..., n$ for a partition of the data set; and $p(a_i)$ is a probability distribution that belongs to the credal set $K(A)$. Credal sets are closed and convex sets of probability distributions [1]. The IIG is applied on credal sets using uncertainty measures of probability distributions [4]. For more details and extended explanations of the IIG see [4, 5, 6]. Different classification trees can be built using the IIG split criterion. For example, one can build a classification tree using the maximum entropy distributions from the credal set of distributions associated with the Imprecise Dirichlet Model (IDM) [1] or with the Nonparametric Predictive Inference for multinomial data (NPI-M) [2], which are introduced in Sections 2.2 and 2.3, respectively. In this paper, we refer to a classification tree built with the IDM by C-IDM *algorithm*, built with NPI-M by C-NPI-M *algorithm* and built with A-NPI-M by C-A-NPI-M *algorithm*, where A-NPI-M stands for Approximate NPI-M. A-NPI-M can be used with the closed and convex set of probability distributions generated by the NPI-M singleton probabilities. A-NPI-M is simpler to use because it is not necessary to consider the set of constraints associated with the NPI-M model.

2.2. Imprecise probabilities

In the middle of the 19th century, the idea of imprecise probabilities was first proposed by Boole [13]. Since then, imprecise probability based methods have been developed for many areas of statistics. An overview of the main aspects of imprecise probabilities theory and applications has been presented by Augustin et al. [9] and by Walley [41].

In classical probability theory, for an event A, a precise probability $p(A) \in [0,1]$ is used to quantify uncertainty about A, where p is a probability measure satisfying Kolmogorov's axioms [9]. In real world data sets, precise probability calculated from the data is likely to be inaccurate, hence, having the possibility to use imprecise probability may give advantages over the use of precise probability. Imprecise probability uses lower and upper probabilities for the event, and hence reflects more uncertainty about the event. Unlike classical probability, in imprecise probability we assign an interval probability for an event A, such as $[\underline{P}(A), \overline{P}(A)]$, where $0 \leq \underline{P}(A) \leq \overline{P}(A) \leq 1$, and where $\underline{P}(A)$ denotes the lower probability and $\overline{P}(A)$ denotes the upper probability for event A. The classical probability is a special case in imprecise probability which occurs when $\underline{P}(A) = \overline{P}(A)$. Complete lack of information about an event A is represented by $\underline{P}(A) = 0$ and $\overline{P}(A) = 1$. Weichselberger [43] defined the structure, $\mathcal{M}$:

$$\mathcal{M} = \{p(.) : \underline{P}(A) \leq p(A) \leq \overline{P}(A), \forall A \in \mathcal{A}\}, \tag{6}$$

where $\mathcal{A}$ is a set of events, and $p(.)$ is a set-function defined on $\mathcal{A}$ satisfying Kolmogorov's axioms in classical probability theory. The lower and upper probabilities for an event A are:

$$\underline{P}(A) = \inf_{p(.) \in \mathcal{M}} p(A) \tag{7}$$

and

$$\overline{P}(A) = \sup_{p(.) \in \mathcal{M}} p(A). \tag{8}$$

In 1996, Walley introduced an imprecise probability model for inference from multinomial data [42], which is called the Imprecise Dirichlet Model (IDM). The IDM is one of the most popular imprecise probability models. Assume that we have a data set with N observations. Let X be a variable whose values, or categories belong to $\{x_1, ..., x_n\}$, and let n_{x_i} denote the total number of observations in x_i, for $i = 1, ..., n$. The IDM-based lower and upper probabilities for the event that the next future observation, X_{n+1} will be in x_i, are

$$\underline{P}_{IDM}(X_{n+1} \in x_i) = \frac{n_{x_i}}{N + \tilde{s}} \tag{9}$$

and

$$\overline{P}_{IDM}(X_{n+1} \in x_i) = \frac{n_{x_i} + \tilde{s}}{N + \tilde{s}}, \tag{10}$$

where $\tilde{s}$ is a parameter which is chosen independently of the data. The value of $\tilde{s}$ determines how quickly the lower and upper probabilities converge when

the sample size increases [42]. Walley suggested to choose the value of the parameter $\tilde{s}$ equal to 1 or 2 [42]. As shown by Abellán [1], the IDM gives imprecise probabilities that lead to the following (closed and convex) credal set of probability distributions,

$$L = \left\{ p \mid p\left(x_i\right) \in \left[\frac{n_{x_i}}{N+\tilde{s}}, \frac{n_{x_i}+\tilde{s}}{N+\tilde{s}} \right], \quad i = 1, \ldots, n, \sum_{i=1}^{n} p(x_i) = 1 \right\}. \qquad (11)$$

The IDM has been applied to many statistical problems in the literature. Some of these applications were reviewed by Bernard [11]. However, the use of the IDM has been criticised for some disadvantages [37]. Some of these disadvantages of the IDM were already discussed by Walley [42], and by other researchers, which motivated researchers to introduce alternative models for inference from multinomial data. Coolen and Augustin [22, 23] proposed a new model for inference from multinomial data, which is Nonparametric Predictive Inference for Multinomial data (NPI-M).

2.3. Nonparametric Predictive Inference (NPI)

Nonparametric Predictive Inference (NPI) is a frequentist statistical method which uses only few model assumptions to learn from the data in the absence of prior knowledge. NPI is based on Hill's assumption $A_{(n)}$ [28], and uses lower and upper probabilities to quantify uncertainty [8]. Hill [28] introduced the assumption $A_{(n)}$ for prediction of future observations when there is no strong prior knowledge about the form of the underlying distribution of a random quantity. Hill's assumption $A_{(n)}$ directly provides probabilities for one or more real-valued future random quantities, based on observed values of related random quantities. Let $X_1, ..., X_n, X_{n+1}$ be real-valued and exchangeable random quantities, where we assume that the probability of ties is zero. Let the ranked observed values of $X_1, ..., X_n$ be denoted by $x_1 < ... < x_n$, and let $x_0 = -\infty$ and $x_{n+1} = \infty$ for ease of notation. These ordered observations partition the real-line into $n+1$ open intervals $I_j = (x_{j-1}, x_j)$ for $j = 1, ..., n+1$. The assumption $A_{(n)}$ states that the next future observation, represented by a random quantity X_{n+1}, is equally likely to fall in any interval I_j with probability $\frac{1}{n+1}$ for each $j = 1, ..., n+1$, i.e. $P(X_{n+1} \in I_j) = \frac{1}{n+1}$. Hill's assumption $A_{(n)}$ does not assume anything else, and it is clearly a post-data assumption related to exchangeability of $n+1$ values on the real-line [25].

NPI has been developed in recent years for different applications in statistics, operations research, finance, risk and reliability [21]. NPI has been presented for different types of data, such as binary data [19], real-valued data [8], right-censored data [24], ordinal data [27] and multinomial data [10, 22, 23]. In Section 2.3.1, we introduce NPI for binary data. Then, we introduce NPI for multinomial data in Section 2.3.2.

2.3.1. NPI for Binary data

This section summarises NPI for binary random quantities as introduced by Coolen [19]. Suppose that we have a sequence of $n+m$ exchangeable binary trials where the possible outcomes of each trial are either 'success' or 'failure', and the data consist of s successes in n trials, and m future trials are considered. Let Y_1^n and Y_{n+1}^{n+m} denote the random number of successes in trials 1 to n, and in trials $n+1$ to $n+m$, respectively.

For a single future observation, i.e. $m = 1$, the NPI lower and upper probabilities are

$$\underline{P}\left(Y_{n+1}^{n+1} = 1 | Y_1^n = s\right) = \frac{s}{n+1} \tag{12}$$

and

$$\overline{P}\left(Y_{n+1}^{n+1} = 1 | Y_1^n = s\right) = \frac{s+1}{n+1}. \tag{13}$$

More details and examples about NPI for binary quantities are given by Coolen [19].

2.3.2. NPI for multinomial data

Coolen and Augustin [8, 22, 23] have developed Nonparametric Predictive Inference for Multinomial data (NPI-M). The NPI-M is based on the *circular-$A_{(n)}$* assumption, which is a variation of Hill's assumption $A_{(n)}$ [22]. Since multinomial data are represented as observations on a probability wheel, and hence as circular data, we use the *circular-$A_{(n)}$* assumption, which is denoted by $\text{Ⓐ}_{(n)}$ [20, 22]. Suppose that we have ordered circular data $y_1 < y_2 < ... < y_n$ which create n intervals on a circle, represented as $I_j = (y_j, y_{j+1})$ for $j = 1, ..., n-1$ and $I_n = (y_n, y_1)$. The assumption $\text{Ⓐ}_{(n)}$ states that the next future observation, represented by a random quantity Y_{n+1}, falls equally likely in any interval I_j for each $j = 1, ..., n$, i.e. $P(Y_{n+1} \in I_j) = \frac{1}{n}$. The $\text{Ⓐ}_{(n)}$ is a post-data assumption related to exchangeability for such circular data.

Coolen and Augustin [23] assume that each observed category is represented by one single segment of the probability wheel, where the segment is an area between two lines from the center to the circumference of the wheel. Combining this assumption with *circular-*$A_{(n)}$ implies that two or more lines representing observations in the same categories are positioned next to each other. Therefore, a slice that is bordered by two lines representing different categories is a separating slice, which could be assigned to any of these different categories or to unobserved category. It is also assumed that there is no ordering of the categories, and hence no ordering of the segments on the wheel.

Coolen and Augustin introduced the NPI-M for the case of a known number of categories [23] and for the case of an unknown number of categories [22]. We restrict our focus in this paper on the case where the number of possible categories, denoted by K, is known. We assume that $K \geq 3$. However, for the case when $K = 2$, the NPI-M can be used, but using NPI for binary data [19] is more appropriate as it leads to slightly less imprecision.

Suppose that there are $K \geq 3$ possible categories denoted by $C_1, ..., C_K$. We assume that $C_1, ..., C_K$ are observed categories. Let n_i represent the number of observations in category C_i for $i = 1, ..., K$, and let the total number of observations be $n = \sum_{i=1}^{K} n_i$.

For events $Y_{n+1} \in C_i$, so considering only a single category, the NPI-M lower and upper probabilities are

$$\underline{P}\left(Y_{n+1} \in C_i\right) = \max\left(0, \frac{n_i - 1}{n}\right) \tag{14}$$

and

$$\overline{P}\left(Y_{n+1} \in C_i\right) = \min\left(\frac{n_i + 1}{n}, 1\right) \tag{15}$$

More details and examples about NPI for Multinomial data are given by Coolen and Augustin [8, 22, 23].

3. Direct NPI classification

In this section, we consider direct classification using NPI for binary data and NPI for multinomial data. The Direct NPI classification method bases classification on the NPI lower and upper probabilities for events containing

binary or multinomial data, without adding any further assumptions. We will first introduce direct NPI classification for binary data in Section 3.1, then direct NPI classification for multinomial data is introduced in Section 3.2.

3.1. Direct NPI classification for binary data

In this section, we introduce Direct NPI classification using NPI for binary data [19], introduced in Section 2.3.1. As a first step to develop the method of Direct NPI classification, we start with exploring the method on completely binary data, where both the class variable and the attribute variables are binary. Assume that we have a data set of n instances which only have two values, 0 or 1. Suppose that there are $T \geq 1$ binary attribute variables. Let A_i indicate attribute variables, for $i \in \{1, ..., T\}$. The value of each attribute is either 0 or 1, i.e. $A_i = 0$ or $A_i = 1$. Let C be a binary class variable, where $C = 0$ or $C = 1$. Let n^0 denote the total number of instances with $C = 0$, and let n^1 denote the total number of instances with $C = 1$, so $n = n^0 + n^1$.

Suppose that we want to see if attribute A_i is useful for indicating the possible class state for a future instance. The attribute A_i is useful if an instance with attribute value $A_i = 1$, has a high probability of being classified as $C = 1$, so that $A_i = 1$ is an indicator for $C = 1$, and an instance with attribute value $A_i = 0$, has a high probability of being classified as $C = 0$, so that $A_i = 0$ is an indicator for $C = 0$. Therefore, we are interested in the conditional events $C = 1|A_i = 1$ and $C = 0|A_i = 0$. Note here that this approach of indication allows the attribute values to be relabelled, possibly multiple times in the construction of a single tree. More clarifications about this issue of relabelling are given in Section 5. Further clarifications and examples are given in [7]. Of course, the ideal situation would be that all instances with attribute value $A_i = 1$ are classified as $C = 1$, and all instances with attribute value $A_i = 0$ are classified as $C = 0$. These conditional events indicate that attribute value 1 ($A_i = 1$) is related to class state 1 ($C = 1$) in terms of the data set, and similarly for $C = 0|A_i = 0$. We consider such events for one future instance for which the attributes are available but we do not know its class states. This instance is assumed to be exchangeable with all other n instances in the data set. Let $n(A_i = 1)$ be the total number of instances in the data set which have value $A_i = 1$. Let $n^1(A_i = 1)$ denote the total number of instances which have value $A_i = 1$ and which are classified as $C = 1$, and let $n^0(A_i = 1)$ denote the total number of instances which have value $A_i = 1$ but are classified as $C = 0$. Thus, $n(A_i = 1) = n^1(A_i = 1) + n^0(A_i = 1)$. Note that these numbers are known from the data set.

It should be emphasized that the inference considers an instance which is not in the data, and hence, its class state is unknown. We denote the unknown class state of one future instance, say 'instance $n+1$', which is not included in the data set, by C_{n+1}. This instance is assumed to be exchangeable with the n other instances in the data set. Judgement on correctness of the predictive inference of this instance is impossible at the time of such predictions, but the effectiveness of such judgments can be considered based on success of the attribute variables for the n available instances in the data set. Using NPI for binary data [19], introduced in Section 2.3.1, see Equations (12) and (13), we can derive the NPI lower and upper probabilities for $C_{n+1} = 1|A_{n+1,i} = 1$. Note here that $A_{n+1,i} = 1$ is the attribute value for this future instance. We can provide the NPI lower and upper probabilities for the event that instance $n+1$ will be classified as $C_{n+1} = 1$ given that its attribute value $A_{n+1,i} = 1$. The NPI lower probability for this event is

$$\underline{P}\left(C_{n+1} = 1|A_{n+1,i} = 1\right) = \frac{n^1\left(A_i = 1\right)}{n\left(A_i = 1\right) + 1}, \tag{16}$$

and the NPI upper probability is

$$\overline{P}\left(C_{n+1} = 1|A_{n+1,i} = 1\right) = \frac{n^1\left(A_i = 1\right) + 1}{n\left(A_i = 1\right) + 1}. \tag{17}$$

Similarly, the NPI lower and upper probabilities for $D_{n+1} = 0|t_{n+1,j} = 0$ are

$$\underline{P}\left(C_{n+1} = 0|A_{n+1,i} = 0\right) = \frac{n^0\left(A_i = 0\right)}{n\left(A_i = 0\right) + 1}, \tag{18}$$

and

$$\overline{P}\left(C_{n+1} = 0|A_{n+1,i} = 0\right) = \frac{n^0\left(A_i = 0\right) + 1}{n\left(A_i = 0\right) + 1}. \tag{19}$$

The NPI lower and upper probabilities for events $C_{n+1} = 0|A_{n+1,i} = 1$ and $C_{n+1} = 1|A_{n+1,i} = 0$ can also be derived via the conjugacy property. For an event E, the conjugacy property is $\overline{P}(E) = 1 - \underline{P}(E^c)$, where E^c is the complementary event to A. It should be noticed that the values of the above NPI lower and upper probabilities will reflect the strength of the evidence for the class state of the future instance, which is not included in the data.

3.2. Direct NPI classification for multinomial data

In this section, we illustrate how we can base classification on the NPI lower and upper probabilities for events with multinomial data, and without adding any further assumptions. Assume that we have a data set of n instances. Suppose that there are $T \geq 1$ attribute variables. Let A_i for $i = \{1, ..., T\}$ indicate these attribute variables, where each attribute variable can have a different number of categories. Let a_{ij} represent the categories in attribute A_i for $i = 1, ..., T$ and $j = 1, ..., h_i$. So, h_i is the number of categories in attribute A_i. Suppose also that we have a target variable with known number of classes. We assume that the target variable is represented by a class variable $C \in \{c_1, ..., c_m\}$. In our Direct NPI classification method we assume that all categories have been observed in the data.

Let n^{c_r} be the number of instances which are classified as class r, for $r = 1, ..., m$, hence, $n = n^{c_1} + n^{c_2} + ... + n^{c_m}$. Let $n(A_i = a_{ij})$ be the total number of instances in the data set which have $A_i = a_{ij}$. Let $n^{c_1}(A_i = a_{ij})$ be the number of instances which have $A_i = a_{ij}$ and which are classified as c_1, so $n(A_i = a_{ij}) = \sum_{r=1}^{m} n^{c_r}(A_i = a_{ij})$ for $r = 1, ..., m$.

Now we will see if attribute A_i is useful or not. Clearly, A_i is useful if there is a high probability that an instance with attribute value $A_i = a_{i1}$ indeed has class c_1, so that $A_i = a_{i1}$ is an indicator for c_1, and an instance with attribute value $A_i = a_{i2}$ indeed has class c_2, so that $A_i = a_{i2}$ is an indicator for c_2, and so on. First, we assume that attribute category a_{i1} is linked with class state c_1, attribute category a_{i2} is linked with class state c_2 and attribute category a_{i3} is linked with class state c_3. It is important to note that this assumption is only considered here to illustrate the main idea of the D-NPI classification method. However, in the experimental analysis in this paper, we link each attribute category with the class state which is most frequently associated with it. Clearly, the number of attribute categories a_{ij} might not be the same as the number of possible states in the class variable c_r. So it is possible for multiple attribute categories to indicate the same class state. With this consideration, we focus on the indication that is given by each category with regard to predicting the possible class state. So, we are interested in the conditional events $(C = c_r | A_i = a_{ij})$.

We consider such events for one future instance for which the attributes values are known but which class status is unknown. Let C_{n+1} denote the unknown class status for a single future instance which is not included in the data. Using NPI for binary data [19], introduced in Section 2.3, we can

provide the NPI lower and upper probabilities for the event that a future instance, which is not included in the data set has class r, c_r given that its attribute value is a_{ij}, $A_{n+1,i} = a_{ij}$, for $j = 1, ..., h_i$, i.e. $C_{n+1} = c_r | A_{n+1,i} = a_{ij}$, for $j = 1, ..., h_i$ and $r = 1, ..., m$. The NPI lower probability is

$$\underline{P}\left(C_{n+1} = c_r | A_{n+1,i} = a_{ij}\right) = \frac{n^{c_r}\left(A_i = a_{ij}\right)}{n\left(A_i = a_{ij}\right) + 1}, \tag{20}$$

and the NPI upper probability is

$$\overline{P}\left(C_{n+1} = c_r | A_{n+1,i} = a_{ij}\right) = \frac{n^{c_r}\left(A_i = a_{ij}\right) + 1}{n\left(A_i = a_{ij}\right) + 1}. \tag{21}$$

We could also use NPI for multinomial data for these lower and upper probabilities, but that would lead to slightly larger imprecision.

The Direct NPI classification for the conditional event $C_{n+1} = c_r | A_{n+1,i} = a_{ij}$ can be calculated via Formulas (20) and (21). It should be noticed that the values of these NPI lower and upper probabilities will directly reflect the strength of the evidence with regard to the possible class state for the single future instance, based on the data. More clarifications and examples are given in [7].

4. Correct Indication

In this section we introduce a novel split criterion to be used when building classification trees based on the Direct NPI classification method. The concept of CI is used to decide on which attribute the data will be split. So, in order to select an attribute variable for each node of the classification tree, the NPI lower and upper probabilities for CI need to be calculated. After that we aim at the largest possible values for both the NPI lower and upper probabilities for CI. The CI reports the strength of the evidence that the attribute variables indicate, based on the data. We introduce the NPI lower and upper probabilities for CI corresponding to binary attribute variables in Section 4.1. Then, the split criterion *Correct Indication* (CI) is generalized to multinomial data in Section 4.2.

4.1. Correct Indication for binary data

In this section, we introduce the NPI lower and upper probabilities for CI corresponding to binary data. Let $p = P(A_i = 1)$, hence, $P(A_i = 0) = 1 - p$. Using NPI for binary quantities [19], introduced in Section 2.3.1, we get

$$p \in \left[\frac{n(A_i = 1)}{n+1}, \frac{n(A_i = 1) + 1}{n+1} \right]. \tag{22}$$

Now we can determine the NPI lower probability for the event that attribute A_i leads to CI by taking the NPI lower probabilities for $(C_{n+1} = 0 | A_{n+1,i} = 0)$, and for $(C_{n+1} = 1 | A_{n+1,i} = 1)$, and p within the range given by (22) to minimise the weighted average,

$$\underline{P}_i(CI) = \min_p \left(\frac{n^0\left(A_i = 0\right)}{n\left(A_i = 0\right) + 1}(1 - p) + \frac{n^1\left(A_i = 1\right)}{n\left(A_i = 1\right) + 1} p \right). \tag{23}$$

This minimum is achieved for

$$p = \begin{cases} \dfrac{n(A_i = 1) + 1}{n+1} & \text{if} \quad \dfrac{n^0(A_i = 0)}{n(A_i = 0) + 1} \geq \dfrac{n^1(A_i = 1)}{n(A_i = 1) + 1}, \\ \dfrac{n(A_i = 1)}{n+1} & \text{otherwise.} \end{cases} \tag{24}$$

Similarly, the NPI upper probability for the event that attribute A_i leads to CI is given by

$$\overline{P}_i(CI) = \max_p \left(\frac{n^0\left(A_i = 0\right) + 1}{n\left(A_i = 0\right) + 1}(1 - p) + \frac{n^1\left(A_i = 1\right) + 1}{n\left(A_i = 1\right) + 1} p \right), \tag{25}$$

where p is such that

$$p = \begin{cases} \dfrac{n(A_i = 1) + 1}{n+1} & \text{if} \quad \dfrac{n^0(A_i = 0) + 1}{n(A_i = 0) + 1} \leq \dfrac{n^1(A_i = 1) + 1}{n(A_i = 1) + 1}, \\ \dfrac{n(A_i = 1)}{n+1} & \text{otherwise.} \end{cases} \tag{26}$$

These NPI lower and upper probabilities for CI should be calculated for each single attribute variable. We aim at the maximum probability for CI for both NPI lower and upper probabilities for a future instance. Generally, in a classification tree, the most informative attribute variable is desired. In CI for binary data, if the attribute variable is very informative in both cases, then it gives high NPI lower and upper probabilities for CI. For example, if all attribute values $A_i = 0$ indicate the class $C = 0$, and all attribute values $A_i = 1$ indicate the class $C = 1$, then this is an example which would give the highest possible NPI lower and upper probabilities for CI.

4.2. Correct Indication for multinomial data

In this section, we generalise the CI formulas to attributes with a known number of categories, $h_i \geq 3$. Recall that h_i is the number of observed categories, labelled $a_{i1}, ..., a_{ih_i}$. In this paper we assume that all possible categories have been observed as generally in classification unobserved categories are never more likely than observed categories. It may also possible to adapt our method to consider unobserved categories in future work.

Let $p_{ij} = P(A_i = a_{ij})$ for $i = 1, ..., T$ and $j = 1, ..., h_i$, where for each attribute variable A_i, $\sum_{i=1}^{h_i} p_{ij} = 1$. Using NPI for multinomial data [23], introduced in Section 2.3, we get

$$p_{ij} \in \left[\frac{n_{ij} - 1}{n}, \frac{n_{ij} + 1}{n} \right] \tag{27}$$

where n_{ij} denotes the number of times we have observed category a_{ij} for A_i.

We can determine the NPI lower probability for the event that attribute A_i leads to CI, by taking the NPI lower probabilities for the events $C_{n+1} = c_1 | A_{n+1,i} = a_{i1}, ..., C_{n+1} = c_m | A_{n+1,i} = a_{ih_i}$, and p_{ij} within the range given by (27) to minimise the weighted average,

$$\underline{P}_i(CI) = \min_{p_{ij} \in \mathcal{P}} \sum_{j=1}^{h_i} \frac{n^{c_r}\left(A_i = a_{ij}\right)}{n\left(A_i = a_{ij}\right) + 1} \, p_{ij} \tag{28}$$

where $\mathcal{P}$ is the set of probability distributions over the categories which correspond to the NPI lower and upper probabilities, and which is defined as follows:

$$\mathcal{P} = \left\{ p \mid \frac{n_{ij} - 1}{n} \leq p_{ij} \leq \frac{n_{ij} + 1}{n}, \forall j = 1, ..., h_i, \sum_{j=1}^{h_i} p_{ij} = 1 \right\}. \tag{29}$$

Similarly, the NPI upper probability for the event that attribute A_i leads to CI is

$$\overline{P}_i(CI) = \max_{p_{ij} \in \mathcal{P}} \sum_{j=1}^{h_i} \frac{n^{c_r}\left(A_i = a_{ij}\right) + 1}{n\left(A_i = a_{ij}\right) + 1} \, p_{ij}. \tag{30}$$

The above NPI lower and upper probabilities for CI should be calculated for each single attribute variable, then we choose the most informative attribute at each stage of building the classification tree based on these NPI lower and upper probabilities.

To compute the NPI lower and upper probabilities for CI, which are given by Equations (28) and (30), we consider all possible configurations δ on the probability wheel (see Section 2.3), applying the *circular-*$A_{(n)}$ assumption to each δ to get corresponding NPI lower and upper probabilities for CI ($\underline{P}_\delta(CI)$ and $\overline{P}_\delta(CI)$), and then we take the NPI lower and upper probabilities with respect to the set $\mathcal{S}$ of all configurations δ such that

$$\underline{P}(CI) = \min_{\delta \in \mathcal{S}} \underline{P}_\delta(CI) \tag{31}$$

and

$$\overline{P}(CI) = \max_{\delta \in \mathcal{S}} \overline{P}_\delta(CI). \tag{32}$$

Next, we derive optimal configurations which lead to the NPI lower and upper probabilities for CI. We first consider the NPI lower probability for CI, then the NPI upper probability for CI is considered.

4.2.1. Lower probability

To find the NPI lower probability for CI for attribute A_i, $\underline{P}_i(CI)$, we need to minimise over all possible configurations of the probability wheel, then choose the configuration that gives the smallest possible value.

Let $f_{ij} = \dfrac{n^{c_r}(A_i = a_{ij})}{n(A_i = a_{ij}) + 1}$, for $i = 1, ..., T$ and $j = 1, ..., h_i$, so we rewrite Equation (28) in the following way

$$\underline{P}_i(CI) = \min_{p_{ij} \in \mathcal{P}} (f_{i1}p_{i1} + f_{i2}p_{i2} + ... + f_{ih_i}p_{ih_i}). \tag{33}$$

Suppose that the fractions f_{ij} are reordered in an increasing way and relabelled such that $\acute{f}_{i1} \leq \acute{f}_{i2} \leq ... \leq \acute{f}_{ih_i}$, with corresponding $\acute{p}_{ij}$. For example, considering attribute variable A_1, if the smallest f_{1j} is f_{12}, then $\acute{f}_{11} = f_{12}$ and $\acute{p}_{11} = p_{12}$. Thus, the NPI lower probability for CI is

$$\underline{P}_i(CI) = \min_{\acute{p}_{ij} \in \mathcal{P}} (\acute{f}_{i1}\acute{p}_{i1} + \acute{f}_{i2}\acute{p}_{i2} + ... + \acute{f}_{ih_i}\acute{p}_{ih_i}). \tag{34}$$

We separate categories corresponding to the largest $\acute{f}_{ij}$ as much as possible to ensure that we can assign probability masses of slices 'in between' to its

neighbour with smaller value of $\acute{f_{ij}}$, for minimisation. Therefore, the configuration of the wheel which gives the most flexibility to do so is the arrangement where categories $a_{i1}, ..., a_{ih_i}$ corresponding to $\acute{p}_{i1}, ..., \acute{p}_{ih_i}$ are permuted in the following way, $a_{i1}, a_{ih_i-1}, a_{i2}, a_{ih_i-2}, a_{i3}, a_{ih_i-3}, ..., a_{ih_i}$.

Generally speaking, to find the NPI lower probability for CI for each attribute variable A_i, each category is assigned its lower probability, $\dfrac{n_{ij}-1}{n}$ and the remaining probability mass is then shared between the categories with the smallest $\acute{f_{ij}}$ in such a way to derive the smallest value of the NPI lower probability for CI. However, the way in which this can be shared must not violate the constraints on the probability wheel for each category. Hence, only $\dfrac{1}{n}$ can be assigned from either side to a category, which implies at most $\dfrac{2}{n}$ in total to any category. We now consider two cases: First, when h_i is even. Then, when h_i is odd.

Case 1: h_i is even. To find the NPI lower probability for CI for each attribute variable A_i, we initially assign probability mass $\dfrac{n_{ij}-1}{n}$ to each category. Once these probability assignments are made, there are h_i separating slices remaining. These separating slices must then be shared equally between the categories with the smallest fractions $\acute{f_{ij}}$, provided that the resulting probabilities are no larger than their upper limits $\dfrac{n_{ij}+1}{n}$. Therefore, the h_i remaining probability masses, each $\dfrac{1}{n}$, must be assigned to a_{i1} to $a_{i\frac{h_i}{2}}$, hence, we assign additional probability mass of $\dfrac{2}{n}$ to each of a_{i1} to $a_{i\frac{h_i}{2}}$.

Example 1. Suppose we have 6 possible categories, a_1, a_2, a_3, a_4, a_5 and a_6, and data where $(n_1, n_2, n_3, n_4, n_5, n_6) = (1, 2, 3, 4, 5, 6)$, so $n = 21$. For simplicity, suppose also that we reorder their corresponding fractions f_{ij}, for $j = 1, ..., 6$ in an increasing order such that $\acute{f_{i1}} \leq ... \leq \acute{f_{i6}}$. Then the NPI lower probability for CI for attribute A_i is

$$\underline{P}_i(CI) = \min_{\acute{p}_{ij} \in \mathcal{P}} (\acute{f_{i1}}\acute{p}_{i1} + \acute{f_{i2}}\acute{p}_{i2} + ... + \acute{f_{i6}}\acute{p}_{i6}) \tag{35}$$

Using NPI for multinomial data, each category is assigned its lower probability $\dfrac{n_{ij}-1}{n}$. Thus, we assign $\left(0, \dfrac{1}{21}, \dfrac{2}{21}, \dfrac{3}{21}, \dfrac{4}{21}, \dfrac{5}{21}\right)$ to $(a_1, a_2, a_3, a_4, a_5, a_6)$,

respectively. After that, there is a total remaining probability mass of $\frac{6}{21}$, which can be assigned to a_1, a_2 and a_3, with the constraint that not more than $\frac{2}{21}$ can be assigned to a single category. The slices separating a_1 from a_5 and a_6 are both assigned to a_1, the slices separating a_2 from a_4 and a_5 are both assigned to a_2, and the slices separating a_3 from a_4 and a_6 are both assigned to a_3. Hence, these assignments are $\left(\frac{2}{21}, \frac{2}{21}, \frac{2}{21}\right)$ to (a_1, a_2, a_3). Therefore, the final assignments are $\left(\frac{2}{21}, \frac{3}{21}, \frac{4}{21}, \frac{3}{21}, \frac{4}{21}, \frac{5}{21}\right)$ to $(a_1, a_2, a_3, a_4, a_5, a_6)$, respectively. Following these assignments we get $\underline{P}_i(CI)$.

□

Case 2: h_i is odd. As for even-valued h_i, we initially assign probability mass $\frac{n_{ij}-1}{n}$ to each category, a_{ij}, for $j = 1, ..., h_i$. Then, as we aim to assign maximal probability masses to the categories with the smallest $\acute{f}_{ij}$, we assign probability mass of $\frac{2}{n}$ to each of a_{i1} to $a_{i(\frac{h_i}{2}-\frac{1}{2})}$. After that we assign the last remaining probability mass $\frac{1}{n}$ to category $a_{i(\frac{h_i}{2}+\frac{1}{2})}$.

Example 2. Consider the same data described in Example 1, excluding the last category a_6 and its corresponding fraction f_{i6}. We then have 5 possible categories where $n = 15$. Recall that their corresponding fractions f_{ij} are reordered, for $j = 1, ..., 5$ in an increasing order. In this example, first, each category is assigned its lower probability $\frac{n_{ij}-1}{n}$. Thus, we assign $\left(0, \frac{1}{15}, \frac{2}{15}, \frac{3}{15}, \frac{4}{15}\right)$ to $(a_1, a_2, a_3, a_4, a_5)$, respectively. Then, we assign the slices separating a_1 from a_4 and a_5 to a_1, and we assign the slices separating a_2 from a_3 and a_4 to a_2. The slice between a_3 and a_5 are assigned to a_3. This means that we assign $\left(\frac{2}{15}, \frac{2}{15}\right)$ to (a_1, a_2), respectively, and we assign the last remaining probability mass $\frac{1}{15}$ to a_3. Therefore, the final assignment is $\left(\frac{2}{15}, \frac{3}{15}, \frac{3}{15}, \frac{3}{15}, \frac{4}{15}\right)$ to $(a_1, a_2, a_3, a_4, a_5)$, respectively, which leads to $\underline{P}_i(CI)$.

□

Following the above method in Case 1 and Case 2 of assigning probability masses to the possible categories, we get the NPI lower probability for CI for attribute A_i, $\underline{P}_i(CI)$.

4.2.2. Upper probability

To find the NPI upper probability for CI for attribute A_i, we need to maximise over all possible configurations on the probability wheel, and then choose a configuration that gives the highest possible value of $\overline{P}_i(CI)$.

Let $f_{ij} = \dfrac{n^{c_r}(A_i = a_{ij}) + 1}{n(A_i = a_{ij}) + 1}$, for $i = 1, ..., T$ and $j = 1, ..., h_i$, so we rewrite the Formula (30) in the following way

$$\overline{P}_i(CI) = \max_{p_{ij} \in \mathcal{P}} (f_{i1}p_{i1} + f_{i2}p_{i2} + ... + f_{ih_i}p_{ih_i}) \tag{36}$$

and then we rearrange the f_{ij} in an increasing order such that $\acute{f}_{i1} \leq \acute{f}_{i2} \leq ... \leq \acute{f}_{ih_i}$. Thus, the NPI upper probability for CI for attribute A_i is

$$\overline{P}_i(CI) = \max_{\acute{p}_{ij} \in \mathcal{P}} (\acute{f}_{i1}\acute{p}_{i1} + \acute{f}_{i2}\acute{p}_{i2} + ... + \acute{f}_{ih_i}\acute{p}_{ih_i}). \tag{37}$$

To maximise these NPI upper probabilities for CI, we separate the largest categories as much as possible to ensure that we can assign probability masses of slices 'in between' to their neighbours with larger value of $\acute{f}_{ij}$. Therefore, we consider the same configuration of the probability wheel that is used to find the NPI lower probability, but we want to assign as much probability mass as possible to the categories with the largest $\acute{f}_{ij}$. Of course, each category is assigned its lower probability $\dfrac{n_{ij} - 1}{n}$, and the remaining probability masses are shared between other categories in such a way to get the largest possible value for the probability for CI. We consider the following two cases to explain how this maximisation can be done.

Case 1: h_i is even. As a first step, we assign probability mass $\dfrac{n_{ij} - 1}{n}$ to each category. As a second step, the remaining probability masses are then shared equally between the categories with the largest possible $\acute{f}_{ij}$. Hence, we assign probability mass of $\dfrac{2}{n}$ to the categories $a_{i(\frac{h_i}{2}+1)}$ to a_{ih_i}. This method leads to the maximum possible value of the NPI upper probability for CI, since we assign the remaining probability masses between the largest categories. Note

that we cannot assign more than probability mass of $\frac{2}{n}$ to any category.

Example 3. Consider the same data described in Example 1. Recall that we reorder their corresponding fractions f_{ij}, for $j = 1, ..., 6$ in an increasing order. First, we assign $\left(0, \frac{1}{21}, \frac{2}{21}, \frac{3}{21}, \frac{4}{21}, \frac{5}{21}\right)$ to $(a_1, a_2, a_3, a_4, a_5, a_6)$, respectively. Then, to derive the NPI upper probability for CI, the slices separating a_6 from a_1 and a_3 are both assigned to a_6, the slices separating a_5 from a_1 and a_2 are both assigned to a_5, and the slices separating a_4 from a_2 and a_3 are both assigned to a_4. These assignments are $\left(\frac{2}{21}, \frac{2}{21}, \frac{2}{21}\right)$ to (a_4, a_5, a_6). Therefore, the final assignments are $\left(0, \frac{1}{21}, \frac{2}{21}, \frac{5}{21}, \frac{6}{21}, \frac{7}{21}\right)$ to $(a_1, a_2, a_3, a_4, a_5, a_6)$, respectively. Following these assignments we get $\overline{P}_i(CI)$.

□

Case 2: h_i is odd. We initially assign probability mass $\frac{n_{ij} - 1}{n}$ to each category, a_{ij}, for $j = 1, ..., h_i$. Then, the best way to distribute the remaining probability masses is to assign probability mass of $\frac{2}{n}$ to the categories $a_{i(\frac{h_i}{2}+\frac{3}{2})}$ to a_{ih_i}. After that, to assign the last probability mass $\frac{1}{n}$ to $a_{i(\frac{h_i}{2}+\frac{1}{2})}$. This distribution of the probability masses leads to the maximum possible value of the probability for CI.

Example 4. Consider the same data in Example 2. Assume also that we reorder their corresponding fractions f_{ij}, for $j = 1, ..., 5$ in an increasing order. To derive the NPI upper probability for CI, we first assign $\left(0, \frac{1}{15}, \frac{2}{15}, \frac{3}{15}, \frac{4}{15}\right)$ to $(a_1, a_2, a_3, a_4, a_5)$, respectively. Then, we assign the slices separating a_5 from a_1 and a_3 to a_5, and we assign the slices separating a_4 from a_1 and a_2 to a_4. The slice between a_2 and a_3 is assigned to a_3. This means that we further assign $\left(\frac{2}{15}, \frac{2}{15}\right)$ to (a_4, a_5), respectively, and $\frac{1}{15}$ to a_3. Therefore, the final assignments are $\left(0, \frac{1}{15}, \frac{3}{15}, \frac{5}{15}, \frac{6}{15}\right)$ to $(a_1, a_2, a_3, a_4, a_5)$, respectively, which leads to $\overline{P}_i(CI)$.

□

Following the above method of assigning probability masses to the possible categories, we get the NPI upper probability for CI for attribute A_i, $\overline{P}_i(CI)$. Finally, we can use these NPI lower and upper probabilities for CI to build classification trees using the D-NPI algorithm, as explained in Section 5.

5. The D-NPI classification tree algorithm

In this section we propose a new algorithm for classification trees which we call *Direct Nonparametric Predictive Inference (D-NPI) classification algorithm*. The building process is similar to the well-known C4.5 algorithm (see Section 2), but we use the CI method introduced in Section 4 as a split criterion to choose the best splitting attribute at each node. Generally, the D-NPI classification tree recursively partitions the training data sets into smaller subsets, based on the most informative attribute variable which is selected by the CI split criterion.

It is important to note that during the process of building D-NPI classification trees in this paper, we assume that $\frac{n^1(A_i=1)}{n(A_i=1)} \geq \frac{n^1(A_i=0)}{n(A_i=0)}$, which means that attribute values are defined such that attribute value 1 (positive value) is related to the class state 1 (positive class state) in terms of the data set. This link between the attribute values and class states should be used at all stages of building the tree, hence, we may need to redefine the attribute values when working with subsets of the data, i.e. at different parts of the tree. Hence, further notation and attention are required.

Suppose that we have a training dataset, $\mathcal{D}$, which has binary or categorical attribute variables, A_i, where $i = 1, ..., T$. Let C be the target variable which also has binary or categorical possible results. The method starts with a tree with a root node, then we go through interior nodes finally arriving at the leaves which describe a possible class state. For the training dataset, $\mathcal{D}$, we first calculate the *Correct Indication* (CI) intervals for the complete list of attribute variables A_i, using the NPI lower and upper probabilities for CI introduced in Section 4 (see Equations (23) and (25) for binary data, and Equations (28) and (30) for multinomial data). Then, we compare the values of the CI intervals for the full training set with the interval given by the NPI lower and upper probabilities for CI if no attribute variable is used, which

are defined in the following paragraph. In this paper, we refer to the NPI lower and upper probabilities for CI if no attribute variable is used by the NPI lower and upper probabilities for NA, where NA is the event of CI in case no attribute variable is used.

Let $\underline{P}(NA)$ denote the NPI lower probability for CI if no attribute variable is used, and let $\overline{P}(NA)$ denote the NPI upper probability for CI if no attribute variable is used. $\underline{P}(NA)$ and $\overline{P}(NA)$ correspond to simply stating the most common value in the target variable. Using the NPI method for binary quantities [19], introduced in Section 2.3.1, see also Equations (12) and (13), the NPI lower and upper probabilities for CI if no attribute variable is used are

$$[\underline{P}(NA), \overline{P}(NA)] = \left[\frac{s}{n+1}, \frac{s+1}{n+1}\right]$$

where s is the number of positive cases in n instances, or the larger value in the target variable C. When the number of positive and negative cases is equal, we choose any of them to calculate the NPI lower and upper probabilities for CI if no attribute variable is used. For the case of multinomial data, the NPI lower and upper probabilities for NA is given by the NPI-M lower and upper probabilities as shown in Section 2.3.2 (see Equations (14) and (15)), which correspond to the largest class state.

After that, for each attribute variable, A_i, we consider whether the NPI lower and upper probabilities for CI for this attribute are greater than the NPI lower and upper probabilities for NA, respectively. That is, the NPI lower probability for CI is greater than the NPI lower probability for NA, and the same for the upper probabilities. If this is the case, then we choose the attribute variable with the highest CI interval as a root node. This means that we consider two conditions in order to split upon the attribute variable with the highest CI values, which are

$$\underline{P}(CI_{i^*}) > \underline{P}(NA) \quad \text{and} \quad \overline{P}(CI_{i^*}) > \overline{P}(NA) \tag{38}$$

where the $\underline{P}(CI_{i^*})$ and $\overline{P}(CI_{i^*})$ correspond to the attribute variable with the highest NPI lower and upper probabilities for CI, and $\underline{P}(NA)$ and $\overline{P}(NA)$ correspond to the highest class state of the target variable. So, i^* indicates the attribute variable that gives the maximum values for the NPI lower and upper probabilities for CI compared to the other attribute variables. If we have two or more attribute variables that all fulfil the two conditions in (38)

but they overlap in their NPI lower and upper probabilities for CI, we choose the one with the highest NPI upper probability for CI. If there are two or more attribute variables that all have the same NPI upper probability for CI and the same NPI lower probability for CI, and they fulfil the two conditions in (38), then we choose any of them to build the tree. If there is no attribute variable that fulfils the two conditions in inequalities (38), we do not split further and transform the node into a leaf with the most common class in the target variable.

The two conditions in (38) are used as a stop criterion when building the D-NPI classification trees. In order to split further when building the D-NPI classification trees, we need to choose attribute variables that produce more information with regard to predicting the possible class state. The NPI lower and upper probabilities for NA are achieved directly from the class variable, so information comes form the class variable only. Therefore, we look for an attribute variable that contains more information than the NPI lower and upper probabilities for NA. An attribute variable that has higher values for the NPI lower and upper probabilities for CI should have more information about predicting the possible class state than only considering the NPI lower and upper probabilities for NA. This stop criterion could prevent overfitting in the D-NPI classification trees, as it prevents us from building larger trees that may overfit the data and these may have less classification accuracy on the testing set.

After selecting the best attribute variable A_{i^*} for the root node, we split the training data set, $\mathcal{D}$, into disjoint subsets $\mathcal{D}_{i^*=a_{i^*j}}$, where $\mathcal{D}_{i^*=a_{i^*j}}$ includes all instances with value $A_{i^*} = a_{i^*j}$, for $j = 1, ..., h_{i^*}$ for the selected attribute variable. For binary data, $\mathcal{D}$ is split into two disjoint subsets $\mathcal{D}_{i^*=0}$ and $\mathcal{D}_{i^*=1}$, where $\mathcal{D}_{i^*=0} \bigcup D_{i^*=1} = \mathcal{D}$ and $\mathcal{D}_{i^*=0} \bigcap \mathcal{D}_{i^*=1} = \emptyset$. While for multinomial data, $\mathcal{D}_{i^*}$ may contain more than two subsets. After this stage, we calculate the CI intervals for each subset and then compare the values of each subset with the corresponding probability interval for NA following the first chosen attribute variable. If there is no attribute variable that satisfies the two conditions in (38), then we do not split further and fix a leaf with the most common class in the target variable. Otherwise, we choose the attribute variable with the highest CI interval to split on, i.e. the NPI lower and upper probabilities for CI. A node will be designated as a leaf node only if we do not have any attribute variable that fulfils the two conditions in (38) or when the observations in the subset all belong to the same class, in which case this class is used as a label to that corresponding leaf node. The D-NPI algorithm continues

Algorithm 1 Pseudocode of the D-NPI algorithm.

Input:
`TR`: Training data set
`Target`: Target variable
`Attr`: List of attribute variables
procedure D-NPI(`TR`, `Target`, `Attr`)
Create a Root node for the tree
    if `TR` have the same class C, **then**
        Return the single-node tree with class C
    if Attr is empty, **then**
        Return the single-node tree with the most common class C in `TR` (*)
    Otherwise
    for each attribute, A in `Attr` **do**
        Compute $\underline{P}_A(CI)$ and $\overline{P}_A(CI)$
        Choose attribute, A with highest $\underline{P}_A(CI)$ and $\overline{P}_A(CI)$
        if $\underline{P}_A(CI) > \underline{P}(NA)$ and $\overline{P}_A(CI) > \overline{P}(NA)$ **then**
            Choose the attribute, A (*)
        else
            Add a leaf node labelled with the most common class in `TR` (*)
    Set A the attribute for Root
    for each value of A, a_j, **do**
        Add a branch below Root, corresponding to $A = a_j$
        Let $\texttt{TR}_{a_j}$ be the subset of `TR` that have $A = a_j$
        if $\texttt{TR}_{a_j}$ is empty, **then**
            Add a leaf node labelled with the most common class in `TR`
        else
            Add the subset generated by D-NPI($\texttt{TR}_{a_j}$, `Target`, `Attr`-$\{A\}$)
return Root
    (*): In case of ties, see descriptions in Section 5.

recursively by splitting further and hence, constructing new subtrees to each branch. Finally, the above process can be represented as a classification tree. Algorithm 1 describes the D-NPI classification tree algorithm, named D-NPI algorithm. Note that this algorithm can be used to build classification trees for binary data or multinomial data, but the split criterion (CI) is differently calculated for each case.

6. Experimental Analysis

In this section, we examine the performance of the D-NPI algorithm on 13 datasets extracted from the UCI repository of machine learning databases [26]. The aim of this experimental analysis is not only to assess the performance

Dataset	N	Att	Range of Att	Classes
Acute Inflammations 1	120	6	2	2
Acute Inflammations 2	120	6	2	2
Banknote authentication	1372	4	2	2
Breast Cancer Wisconsin	699	9	2	2
Congressional Voting Records	435	16	2	2
CMC	1473	9	2-4	3
Hayes-Roth	160	5	3-4	3
Lenses	24	4	2-3	3
Modified Iris	150	4	3	3
Monk's Problems-1	124	7	2-4	2
Nursery	12960	8	2-5	5
Post-Operative Patient	90	8	2-4	3
Qualitative-Bankruptcy	250	6	3	2

Table 1: Datasets description.

of the D-NPI algorithm, but also to compare it with the commonly used C4.5 algorithm and with algorithms based on imprecise probabilities, namely the C-NPI-M, the C-A-NPI-M and the C-IDM (with two choices of the parameter $\tilde{s}$). These algorithms are introduced in Section 2.

6.1. Experimental setup

The experiments were conducted using the statistical R software. The datasets used in this experimental analysis are diverse in terms of their size, the number of classes and the range of categories of the attribute variables. A summary of the main characteristics of each data set is given in Table 1. A brief description of each dataset is provided below to ensure the analysis is self-contained and informative for readers. Additional details about these datasets can be found in [26].

- Acute Inflammations: This medical dataset was created by a domain expert to evaluate an expert system designed for the presumptive diagnosis of two urinary system diseases: acute bladder inflammation and acute nephritis. The dataset is used in two separate classification tasks, with each disease serving as the class variable in its respective version.

- Banknote authentication: This dataset was created from images taken to assess a banknote authentication procedure. The images consist of both authentic and counterfeit banknote-like specimens, digitized using an industrial camera commonly utilized in print inspection. The high-resolution grayscale images were processed using Wavelet Transform techniques to extract numerical features.

- Breast Cancer Wisconsin: This dataset comes from clinical cases reported by Dr. Wolberg over several years, reflecting a chronological grouping of samples. It was gathered as part of a diagnostic study for breast cancer and contains attributes derived from digitized images of fine-needle aspirates of breast masses. Each case is classified as either benign or malignant. The dataset has been revised to correct or eliminate incomplete entries and is commonly used for binary classification tasks in medical diagnostics.

- Congressional Voting Records: This dataset contains the voting records from the 1984 session of the United States House of Representatives. Each entry represents a congressperson classified as either a Republican or a Democrat. The data contains their positions on sixteen fundamental issues identified by the Congressional Quarterly Almanac (CQA). Vote types are grouped into simple categories: yea, nay, or unknown, based on different expressions of support, opposition, or abstention. This dataset is used for a political classification problem.

- Contraceptive Method Choice (CMC): This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey, focusing on health and medicine. It includes responses from married women who were either not pregnant or unsure about their pregnancy status during the interview. The classification task is to predict a woman's contraceptive method choice—no use, long-term, or short-term—based on her demographic and socio-economic attributes.

- Hayes-Roth: This dataset originates from a human subjects study in the social sciences, focused on categorization tasks. It includes several numeric-valued attributes, of which only a subset—age, educational status, and marital status—are used during testing.

- Lenses: This dataset was created to study the challenges associated with fitting contact lenses. It is a clean dataset where each instance is complete and accurate, covering all possible combinations of attribute-value pairs. While the dataset provides a clean and structured environment for classification, it fails to capture the full complexity of real-world decision-making in lens prescriptions. The task involves classifying patients based on limited attributes for lens recommendations.

- Modified Iris: This is a classic dataset in the field of biology and one of the earliest examples used to assess classification methods, initially

introduced by Fisher in 1936. Each instance represents a sample from an iris plant, classified into one of three species. In our version, the continuous attributes have been discretized into categorical values to align with the requirements of our classification approach. The dataset is a widely recognized benchmark in the fields of statistics and machine learning.

- Monk's Problems-1: This dataset includes three artificial classification tasks within a common attribute space to evaluate the performance of different learning algorithms. Each MONK problem, developed for an international comparison of classification techniques, presents a unique logical concept to be learned, with one of them incorporating additional noise. The MONK-1 problem defines its concept as instances where the first and second attributes are equal, or the fifth attribute equals one.

- Nursery: This dataset comes from a hierarchical decision model created to rank applications for nursery schools in Ljubljana, Slovenia, during a time of high enrollment demand in the 1980s. The decision-making process considered various socio-economic and health factors, such as parental occupation, family structure, financial status, and social and health conditions. The original model included intermediate hierarchical concepts, but this dataset simplifies the structure by linking the final decision directly to eight input attributes. This is especially useful for assessing classification algorithms, particularly those related to constructive induction or structure discovery.

- Post-Operative Patient: This dataset is from the health and medicine field, focusing on post-surgical management of patients. The classification task is determining the appropriate next location for patients in a post-operative recovery area. Since hypothermia is a critical concern after surgery, the input attributes mainly reflect various body temperature indicators.

- Qualitative-Bankruptcy: This dataset from the field of computer science aims to predict bankruptcy based on qualitative assessments provided by domain experts. The evaluation encompasses six distinct categorical attributes about various business risk dimensions. These include industrial risk, management risk, and financial flexibility. Each attribute is meticulously evaluated and classified as positive, average, or negative, providing a clear picture of a business's potential challenges and strengths. The classification task assesses whether a firm is likely to go bankrupt. The

dataset originates from expert-generated decision rules and was initially introduced in a study that utilized genetic algorithms to identify classification patterns in qualitative bankruptcy data.

As a first step of developing the D-NPI algorithm, we apply the algorithm on only binary datasets, where both the class variable and attributes are binary. The first five datasets in Table 1 are used for this analysis. As we consider only binary datasets at this step, all continuous attribute variables are converted to binary ones using the same thresholds given by the Information Gain Ratio criterion [39]. After that, all classification tree algorithms are built on these datasets. Note that the Acute Inflammations data set has two target variables, based on each we construct a tree. Thus, we consider it as two separate data sets, each one with a different target variable. To construct the D-NPI algorithm for the first five datasets in Table 1, we use the CI split criterion presented in Section 4.1. On the other hand, the CI split criterion for multinomial datasets that is presented in Section 4.2 is used for the rest of the datasets.

The Nursery data set is large, so to reduce the amount of computation required for this data set, we fix a minimum split number of 100 observations. A minimum split value is sometimes fixed to reduce the required computation as done by Bertsimas and Dunn [12]. In the modified Iris data set, we convert four continuous attributes to categorical attributes with three categories coded as, 'L', 'M' and 'H'. All other datasets only have categorical attributes. All missing values were replaced with modal values. Finally, all classification algorithms have been applied to all these datasets under the same circumstances of pre-analysis steps to ensure a fair comparison.

Six classification algorithms have been used in this analysis, which are the D-NPI, the C4.5, the C-NPI-M, the C-A-NPI-M and the C-IDM with $\tilde{s} = 1$ and $\tilde{s} = 2$. We denote the C-IDM with $\tilde{s} = 1$ and $\tilde{s} = 2$ by C-IDM1 and C-IDM2, respectively. A 10-fold cross-validation procedure has been used for each dataset, then the average results are reported. Classification accuracy rates are used to measure and compare the performance of each classifier. It is the most commonly used method to measure the performance of classification algorithms. It is calculated as the ratio of the total number of correctly classified instances on the testing set to the total number of instances in the testing set. For further analysis of the D-NPI algorithm and to compare it with other algorithms, we used in-sample accuracy which is the classification

accuracy rate on the training set [12, 36]. The in-sample accuracy measure is not commonly used to indicate classification accuracy, but it gives insight into how the algorithm performs on the training set. If the classification algorithm performs very well on the training set but not very well on the testing set, this is likely to indicate overfitting. Thus, the in-sample accuracy is reported to show the performance of classification algorithms on the training set and to check on possible overfitting.

6.2. Results

First, the performance of the D-NPI algorithm has been evaluated against the five other algorithms. Table 2 presents the classification accuracies of the proposed D-NPI algorithm and all other algorithms for each dataset. The results in Table 2 indicate that D-NPI slightly outperforms the other algorithms in 9 datasets. However, for the Acute Inflammations 1 data set, the D-NPI algorithm does not achieve the complete classification accuracy rate demonstrated by the other algorithms, but the D-NPI algorithm produces relatively smaller trees than the other algorithms for this data set. Note that this data set is created by a medical expert as a data set to test the expert system, which will perform the presumptive diagnosis of two diseases of the urinary system, hence it is not a strange situation to have a full classification accuracy rate by different algorithms. Some other researchers have also used this data set and have got a full classification accuracy rate, see Kadhem and Zeki [29] and Medjahed et al. [32], but Chandra and Bhaskar [14] and by Bertsimas and Dunn [12] have reported lower classification accuracy for this data set. For the Banknote authentication, Breast Cancer Wisconsin and Congressional Voting Records datasets, the D-NPI algorithm performs slightly better than the other algorithms, although all classifications algorithms have a very similar classification accuracy results. The D-NPI algorithm returns relatively smaller trees than the other algorithms when binary datasets are used.

For the Lenses data set, there is a clear difference in the classification accuracies among these algorithms, where the D-NPI algorithm clearly performs better than the C4.5 and C-IDM1 algorithms, and performs slightly better than the C-NPI-M, C-A-NPI-M and C-IDM2 algorithms. The reduction in the accuracy for the C4.5 and C-IDM1 algorithms might be because they sometimes stop earlier than other algorithms when building classification trees. However, the clear difference in the classification accuracy between different algorithms could also be because it is a small data set with only 24 instances. For the Monk's Problem-1 dataset, the D-NPI algorithm has the

Dataset	D-NPI	C4.5	C-NPI-M	C-A-NPI-M	C-IDM1	C-IDM2
Acute Inflammations 1	94.17	**100**	**100**	**100**	**100**	**100**
Acute Inflammations 2	**100**	**100**	**100**	**100**	**100**	**100**
Banknote authentication	**89.50**	89.49	89.49	89.49	89.49	89.49
Breast Cancer Wisconsin	**94.85**	94.27	93.19	93.19	93.19	93.19
Congressional Voting Records	**95.64**	95.58	95.58	95.58	95.58	95.35
CMC	**45.49**	45.31	42.93	42.93	42.93	42.93
Hayes-Roth	66.76	66.92	64.62	64.62	63.08	**67.69**
Lenses	**81.67**	70.00	80.00	80.00	75.00	80.00
Modified Iris	**95.33**	92.67	90.00	90.00	92.67	92.67
Monk's Problems-1	**73.33**	69.17	69.17	69.17	69.17	69.17
Nursery	**90.37**	89.21	89.20	89.20	89.20	89.20
Post-Operative Patient	67.78	68.89	**71.11**	**71.11**	**71.11**	**71.11**
Qualitative-Bankruptcy	**99.60**	98.00	98.40	98.40	98.40	98.40
Average	**84.19**	83.04	83.36	83.36	83.07	83.79

Table 2: Classification accuracy results for classification algorithms built with 10-fold cross validation.

highest classification accuracy of 73.33%, where all other algorithms have the same accuracy of 69.17% but with different trees generated by these algorithms. For this data set, D-NPI returns relatively larger trees than the other algorithms which might be the reason for its better performance. For the Hayes-Roth data set, the C-IDM2 algorithm performs slightly better than the other algorithms with classification accuracy of 67.69%. For the CMC, Modified Iris, Nursery and Qualitative-Bankruptcy data sets, the D-NPI algorithm performs better than the other algorithms. Overall, according to the average classification accuracy rate, we can say that all algorithms are performing similarly, but with a slightly better performance by the D-NPI algorithm.

Secondly, following [12, 36], we have used the in-sample accuracy rate to measure the performance of the D-NPI algorithm on the training set, and to compare it with the other algorithms. It is known that if the classification algorithm performs very well on the training set but not very well on the testing set, this indicates likely overfitting. Thus, the in-sample accuracy is reported to show the performance of algorithms on both training and testing sets. Table 3 shows the in-sample accuracy results of classification algorithms. The D-NPI algorithm performs slightly better than the other algorithms in several datasets, followed by the C4.5 algorithm which is also performs better compared to other algorithms in some datasets. It is noticed that the C-IDM2 algorithm performs slightly better than the C-IDM1 algorithm with regard to both average classification accuracy and average in-sample accuracy rates. In this experimental analysis, the C-NPI-M and C-A-NPI-M algorithms are equivalent in all performance measures. These two algorithms do not always lead to the same result as shown by Baker [10]. Finally, the D-NPI algorithm

Dataset	D-NPI	C4.5	C-NPI-M	C-A-NPI-M	C-IDM1	C-IDM2
Acute Inflammations 1	94.17	**100**	**100**	**100**	**100**	**100**
Acute Inflammations 2	**100**	**100**	**100**	**100**	**100**	**100**
Banknote authentication	**89.51**	**89.51**	**89.51**	**89.51**	**89.51**	**89.51**
Breast Cancer Wisconsin	**95.31**	94.20	93.67	93.67	93.84	93.67
Congressional Voting Records	95.63	**95.64**	**95.64**	**95.64**	**95.64**	**95.64**
CMC	**48.84**	47.22	45.17	45.17	45.17	45.17
Hayes-Roth	**83.67**	81.51	79.33	79.33	81.51	82.35
Lenses	**87.49**	84.55	85.91	85.91	85.00	85.91
Modified Iris	95.33	**95.41**	92.07	92.07	94.07	94.07
Monk's Problems-1	**84.24**	74.82	73.66	73.66	73.66	73.66
Nursery	**90.37**	89.21	89.26	89.26	89.26	89.26
Post-Operative Patient	71.23	**71.36**	71.11	71.11	71.11	71.11
Qualitative-Bankruptcy	**99.60**	99.24	98.40	98.40	98.40	98.40
Average	**87.34**	86.36	85.67	85.67	85.93	86.06

Table 3: In-sample accuracy results for classification algorithms built with 10-fold cross validation.

has the highest average result of in-sample accuracy compared to the other algorithms. It should be clarified that the D-NPI algorithm has good results on in-sample accuracy and classification accuracy as well, which may indicate that it does not suffer from overfitting. For example, the D-NPI algorithm has a largest in-sample accuracy rate in the Monk's Problems-1 data set compared to other algorithms, but it also has the largest classification accuracy rate on testing set.

Thirdly, in order to compare different trees generated by the classification algorithms, the average tree size for each algorithm is reported. Table 4 shows the average tree size for each algorithm. Note that we refer to tree size as the total number of leaf nodes, as was done by Bertsimas and Dunn [12], and Murthy and Salzberg [36]. However, other researchers may consider the total number of all nodes. It can be observed from Table 4 that the average tree size of all algorithms is nearly equivalent with some smaller trees generated by the C4.5 algorithm followed by the D-NPI algorithm. From the experimental analysis that is done on all these datasets, it is noticed that the D-NPI algorithm generates relatively smaller trees than other algorithms when applied to binary datasets, but it does not have the smallest trees for multinomial datasets. With regard to the C-IDM1 and C-IDM2 algorithms, we notice that increasing the value of the parameter $\tilde{s}$ could lead to generating smaller trees. This result for the size of trees generated by the C-IDM algorithms has also been reported by Abellán et al. [3] in an extensive experiment to assess the performance of different classification tree algorithms.

Algorithm	D-NPI	C4.5	C-NPI-M	C-A-NPI-M	C-IDM1	C-IDM2
Average	6.48	**5.98**	7.17	7.17	7.64	6.93

Table 4: Average tree size for classification algorithms built with 10-fold cross validation.

To summarise, from the classification accuracy given in Table 2, we draw the following conclusions about the performance of the D-NPI algorithm. The D-NPI algorithm is performing well and slightly better than the other algorithms. The C-NPI-M and C-A-NPI-M algorithms are performing the same on all these data sets. The C-IDM2 algorithm performs better than the C-IDM1 algorithm with regard to this measure. With regard to the in-sample accuracy, the D-NPI algorithm slightly performs better compared to other algorithms followed by the C4.5 algorithm. As the D-NPI algorithm performs well with regard to both the classification accuracy and in-sample accuracy, this could be an indication it does not overfit the data sets. Finally, the C4.5 algorithm has the smallest average tree size, while the C-IDM1 algorithm has the largest average tree size.

7. Concluding remarks

In this paper, we have proposed a new algorithm to build classification trees from Nonparametric Predictive Inference perspective, which we call the Direct Nonparametric Predictive Inference (D-NPI) algorithm. The D-NPI classification algorithm uses a new split criterion, Correct Indication (CI), which is completely based on the lower and upper probabilities given by NPI for binary or multinomial data and it does not use any other assumptions or any added concepts such as entropy. The NPI lower and upper probabilities for CI are also used as a stopping criterion, by comparing them with the NPI lower and upper probabilities for CI in case no further attribute variable is used. The performance of the D-NPI algorithm has been tested against different classification tree algorithms using different performance measures on many datasets from the UCI repository of machine learning. It has been shown that the D-NPI algorithm slightly performs better than the other algorithms.

There are many interesting topics for future research related to this paper. It will be of interest to develop the D-NPI approach for real-valued data. It will also be interesting to explore the use of the D-NPI classification algorithm in random forests. This paper focused on developing and evaluating the D-NPI algorithm based on single-attribute splits. Future work will explore

extending the D-NPI to account for interactions between attributes, aiming to improve performance in more complex, high-dimensional settings while maintaining the interpretability and minimal-assumption philosophy of the NPI framework. Another interesting extension to this work is to develop the D-NPI algorithm with imprecise classification, which might return a set of states rather than the most single state in the class variable. Further future work is to use CI in combination with other inference methods for building classification trees such as Imprecise Dirichlet Model (IDM). It is also interesting to consider developing the D-NPI classification approach with taking the cost of misclassification into account. Similar to this consideration, Moral-García et al. [35] have developed a cost-sensitive classification tree model based on NPI. It will also be interesting to study the stop criterion given in this paper in more detail. For example, using the imprecision (i.e. the difference between the NPI lower and upper probabilities for CI, and for NA) along with the two conditions in (38) particularly when there is overlap between two or more attribute variables that all fulfil these conditions or in other cases when one condition of (38) is met but not the another one. Another work which is currently under investigation is measuring the performance of the D-NPI algorithm when it is applied to noisy data.

R code

The R code is available from the first author on request.

References

[1] Abellán, J. (2006). Uncertainty measures on probability intervals from the imprecise Dirichlet model. *International Journal of General Systems*, 35:509–528.

[2] Abellán, J., Baker, R. M., and Coolen, F. P. A. (2011). Maximising entropy on the nonparametric predictive inference model for multinomial data. *European Journal of Operational Research*, 212:112–122.

[3] Abellán, J., Baker, R. M., Coolen, F. P. A., Crossman, R. J., and Masegosa, A. R. (2014). Classification with decision trees from a nonparametric predictive inference perspective. *Computational Statistics and Data Analysis*, 71:789–802.

[4] Abellán, J. and Masegosa, A. R. (2010). An ensemble method using credal decision trees. *European Journal of Operational Research*, 205:218–226.

[5] Abellán, J. and Moral, S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18:1215–1225.

[6] Abellán, J. and Moral, S. (2005). Upper entropy of credal sets. applications to credal classification. *International Journal of Approximate Reasoning*, 39:235–255.

[7] Alharbi, A. A. H. (2022). *Direct Nonparametric Predictive Inference Classification Trees*. PhD thesis, Durham University. `https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.857563`.

[8] Augustin, T. and Coolen, F. P. A. (2004). Nonparametric predictive inference and interval probability. *Journal of Statistical Planning and Inference*, 124:251–272.

[9] Augustin, T., Coolen, F. P. A., De Cooman, G., and Troffaes, M. C. (2014). *Introduction to Imprecise Probabilities*. Wiley, Chichester.

[10] Baker, R. M. (2010). *Multinomial Nonparametric Predictive Inference: Selection, Classification and Subcategory Data*. PhD thesis, Durham University. `https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.512919`.

[11] Bernard, J.-M. (2005). An introduction to the imprecise dirichlet model for multinomial data. *International Journal of Approximate Reasoning*, 39:123–150.

[12] Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106:1039–1082.

[13] Boole, G. (1854). *An Investigation of the Laws of Thought on which are founded the Mathematical Theories of Logic and Probabilities*. Walton and Maberly, London.

[14] Chandra, B. and Bhaskar, S. (2011). A new approach for classification of patterns having categorical attributes. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, Anchorage, AK, USA. pp. 960-964.

[15] Chang, M., Coolen, F. P., Coolen-Maturi, T., and Huang, X. (2024). A generalized system reliability model based on survival signature and multiple competing failure processes. *Journal of Computational and Applied Mathematics*, 435:115364.

[16] Chang, M., Huang, X., Coolen, F. P., and Coolen-Maturi, T. (2023). New reliability model for complex systems based on stochastic processes and survival signature. *European Journal of Operational Research*, 309(3):1349–1364.

[17] Changala, R., Gummadi, A., Yedukondalu, G., and Raju, U. (2012). Classification by decision tree induction algorithm to learn decision trees from the class-labeled training tuples. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2:427–434.

[18] Coolen, F. P. and Coolen-Maturi, T. (2024). Survival signature for reliability quantification of large systems and networks. In *International Conference on Dependability of Computer Systems*, pages 29–37. Springer.

[19] Coolen, F. P. A. (1998). Low structure imprecise predictive inference for bayes' problem. *Statistics and Probability Letters*, 36:349–357.

[20] Coolen, F. P. A. (2006). On nonparametric predictive inference and objective bayesianism. *Journal of Logic, Language and Information*, 15:21–47.

[21] Coolen, F. P. A. (2011). Nonparametric predictive inference. *International Encyclopedia of Statistical Science*. Springer, pp. 968-970.

[22] Coolen, F. P. A. and Augustin, T. (2005). Learning from multinomial data: a nonparametric predictive alternative to the imprecise Dirichlet model. In *International Symposium on Imprecise Probability: Theories and Applications*, Pittsburgh, Pennsylvania. pp. 125-134.

[23] Coolen, F. P. A. and Augustin, T. (2009). A nonparametric predictive alternative to the imprecise dirichlet model: the case of a known number of categories. *International Journal of Approximate Reasoning*, 50:217–230.

[24] Coolen, F. P. A. and Yan, K. (2004). Nonparametric predictive inference with right-censored data. *Journal of Statistical Planning and Inference*, 126:25–54.

[25] De Finetti, B. (1974). *Theory of Probability*. Wiley, London.

[26] Dua, D. and Graff, C. (2019). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. `http://archive.ics.uci.edu/ml`.

[27] Elkhafifi, F. F. and Coolen, F. P. A. (2012). Nonparametric predictive inference for accuracy of ordinal diagnostic tests. *Journal of Statistical Theory and Practice*, 6:681–697.

[28] Hill, B. M. (1968). Posterior distribution of percentiles: Bayes' theorem for sampling from a population. *Journal of the American Statistical Association*, 63:677–691.

[29] Kadhem, M. H. and Zeki, A. M. (2014). Prediction of urinary system disease diagnosis: A comparative study of three decision tree algorithms. In *2014 IEEE International Conference on Computer Assisted System in Health*, Kuala Lumpur, Malaysia. pp. 58-61.

[30] Maimon, O. Z. and Rokach, L. (2014). *Data Mining with Decision Trees: Theory and Applications*. World Scientific, Singapore.

[31] Maturi, T. A., Coolen-Schrijner, P., and Coolen, F. P. A. (2009). Nonparametric predictive pairwise comparison for real-valued data with terminated tails. *International Journal of Approximate Reasoning*, 51:141–150.

[32] Medjahed, S. A., Saadi, T. A., and Benyettou, A. (2015). Urinary system diseases diagnosis using machine learning techniques. *International Journal of Intelligent Systems and Applications*, 5:1–7.

[33] Moral, S., Mantas, C. J., Castellano, J. G., and Abellán, J. (2020). Imprecise classification with non-parametric predictive inference. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, Cham. pp. 53-66.

[34] Moral-García, S., Mantas, C. J., Castellano, J. G., and Abellán, J. (2020). Non-parametric predictive inference for solving multi-label classification. *Applied Soft Computing*, 88:106011.

[35] Moral-García, S., Abellán, J., Coolen-Maturi, T., and Coolen, F. P. A. (2022). A cost-sensitive imprecise credal decision tree based on nonparametric predictive inference. *Applied Soft Computing*, 123:108916.

[36] Murthy, S. K. and Salzberg, S. (1995). Decision tree induction: How effective is the greedy heuristic? In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Montreal. pp. 222-227.

[37] Piatti, A., Zaffalon, M., and Trojani, F. (2005). Limits of learning from imperfect observations under prior ignorance: the case of the imprecise Dirichlet model. In *International Symposium on Imprecise Probability: Theories and Applications*, Pittsburgh, Pennsylvania. pp. 276-286.

[38] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

[39] Quinlan, J. R. (1993). C4.5: programs for machine learning. *Morgan Kuafmann*.

[40] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

[41] Walley, P. (1991). *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London.

[42] Walley, P. (1996). Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society: Series B*, 58:3–34.

[43] Weichselberger, K. (2000). The theory of interval-probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, 24:149–170.