Entropy based classification trees

Paul Fink
Ludwig–Maximilians–Universität, Munich
paul.fink@stat.uni-muenchen.de

Richard J Crossman
University of Warwick
r.j.crossman@warwick.ac.uk

Abstract

One method for building classification trees is to choose split variables by maximising expected entropy. This can be extended through the application of imprecise probability by replacing instances of expected entropy with the maximum possible expected entropy over credal sets of probability distributions. Such methods may not take full advantage of the opportunities offered by imprecise probability theory. In this paper, we change focus from maximum possible expected entropy to the full range of expected entropy. We present an entropy minimisation algorithm using the non–parametric inference approach to multinomial data. We also present an interval comparison method based on two user–chosen parameters, which includes previously presented splitting criteria (maximum entropy and entropy interval dominance) as special cases. This method is then applied to 13 datasets, and the various possible values of the two user–chosen criteria are compared with regard to each other, and to the entropy maximisation criteria which our approach generalises.

Keywords. Imprecise probability, classification trees, nonparametric predictive inference

1 Introduction

The process of classification involves the splitting of a heterogeneous data space into homogeneous disjoint subspaces with respect to the nominal class(ification) variable C, with the aim of predicting future values of C. This is achieved by determining the splits through the values of feature/attribute variables $(X_1, \ldots, X_n)$. Let C take values/categories in $\mathcal{C} = \{c_1, \ldots, c_K\}$ and each X_i take values in the corresponding set $\mathcal{X}_i$, where for reasons of simplicity the feature variables are assumed to be on a nominal scale. The key consideration is how the homogeneous subspaces are to be constructed.

One method is a classification tree, which partitions the data space into orthotope shaped subspaces. The tree is grown from the root node, which corresponds to the complete data set, and ends in disjoint subsets known as leaves; this is done by recursively applying a splitting procedure. In this paper we consider only k–array splitting as in [4] which is based on Quinlan's ID3 [12] algorithm. In each step an optimal split variable with respect to an impurity criterion is evaluated, which is then assigned to the node; the data contained in the node are then split according to the values of this split variable. If no such optimal split variable may be found the node is declared as a leaf. A value of C is assigned to each leaf, this value is the most frequent category in its corresponding data subset (in the case of a tie, the most frequent category in the data subset of its parent node is used, and so on).

The optimality of a split candidate within a node is measured by the gain in a pre–specified information measure IM. Let N be the data relevant to the node. The information criterion for the node, $IM(N)$, and for each of the unassigned attribute variables X_i, $IM(N|X_i)$ (the information criterion evaluated following a split in X_i of N according to the values of X_i), are then calculated. A split is performed if $IM(N) < IM(N|X_i)$ for some X_i.

A reasonable measure is the *Information Gain*, based on Shannon's entropy [13]. Define $n^N = |N|$ and n_j^N the number of instances within N of class c_j, and furthermore denote the relative frequencies

$$p_j^N = \frac{n_j^N}{n^N},\ p_j^{\hat{x}_i} = \frac{n_j^{\hat{x}_i}}{n^{\hat{x}_i}}, \tag{1.1}$$

with $\hat{x}_i = \{\boldsymbol{d} \in N | X_i = x_i\}$, then the information of N following a split in X_i is defined as

$$I(N, X_i) = \sum_{x_i \in \mathcal{X}_i} p(X_i = x_i) H(\boldsymbol{p}^{\hat{x}_i}), \tag{1.2}$$

where $p(X_i = x_i)$ is also estimated by relative fre-

quencies and $H(\cdot)$ is the Shannon–Entropy defined as

$$H(\boldsymbol{p}) = -\sum_{j=1}^{K} p_j \ln(p_j), \tag{1.3}$$

for probability distribution $\boldsymbol{p}$. $H(\boldsymbol{p})$ attains its minimum (0) for some $p_j = 1$ and its maximum ($\ln(K)$) for the uniform distribution. While the probability distribution attaining the maximum is unique for fixed K, this does obviously not hold for the one attaining the minimum.

Finally, the Information Gain is defined as

$$IM(N, X_i) = H(\boldsymbol{p}^N) - I(N, X_i). \tag{1.4}$$

In determining the split variable only $I(N, X_i)$ in (1.4) is relevant. Maximising (1.4) implies minimising $I(N, X_i)$ which requires minimising entropy.

Up to this point the probabilities $p_j^{\cdot} = P(C = c_j|\cdot)$ were estimated by classical relative frequencies and thus too is the associated probability distribution. In [4] this single distribution is replaced by a credal set of probability distributions estimated by the Imprecise Dirichlet Model (IDM), giving intervals for $p_j^{\cdot}$ of

$$p_j^{\hat{x}_i} \in \left[\frac{n_j^{\hat{x}_i}}{n^{\hat{x}_i} + s}, \frac{n_j^{\hat{x}_i} + s}{n^{\hat{x}_i} + s} \right]. \tag{1.5}$$

Note that s influences the degree of imprecision; this parameter is commonly set to $s = 1$ or $s = 2$.

There are alternatives to the IDM; the Non-Parametric Predictive Inference (NPI) approach [6] is one. This is applied in [5] and [8] to replace the IDM with the multinomial NPI and ordinal NPI, respectively. A short introduction to this method follows.

The NPI approach is is designed to assume as little as is possible about a distribution from which observations are taken. Assume n observations $x_1, \ldots, x_n$ have been made. In the ordinal case, these are relabelled so that $x_1 < x_2 < \ldots x_n$. It is then assumed that observation x_{n+1} has probability $\frac{1}{n+1}$ of being smaller than x_1, the same probability of being larger than x_n, and the same probability of lying in any given data interval $I_{j+1} = [x_j, x_{j+1}]$ for $1 \leq j \leq n-1$ (we set $I_1 = (-\infty, x_1]$ and $I_{n+1} = [x_n, \infty)$). This is known as Hill's assumption, $A_{(n)}$ [11].

By using a latent variable approach, a category c_j in $\mathcal{C}$ can be considered as equivalent to some interval IC_j overlapping the data intervals. The interval IC_j itself is unknown (though IC_1 and IC_K have known bounds at negative and positive infinity, respectively), but its bounds must lie within data intervals which have an observation c_j as exactly one bound. Therefore each interval I_k can be said to be either entirely within IC_j, partially within it, or wholly outside it. The lower probability that $x_{n+1} \in c_j$ is then simply calculated by summing the probability mass of all intervals I_k which lie entirely within IC_j. The upper probability that $x_{n+1} \in c_j$ is calculated by summing the probability mass of all intervals I_k with a non-zero intersection with IC_j.

In the case of multinomial data, these intervals are represented as slices on a probability "wheel", with the observations that forming the interval boundaries representing the lines separating those slices. Observation x_{n+1} has equal chance $\frac{1}{n}$ of falling within any given slice on the wheel. This is referred to as the circular Hill assumption, or circular-$A_{(n)}$.

All observations of the same category are adjacent on the wheel, and any slices between those observations must be assigned to that category. Slices between two different observations can be assigned to either or both those observations, and/or to a previously unobserved condition (since slices for a given category are adjacent, a given unobserved category can be assigned to at most one such slice).

Therefore the lower probability of category j is equal to the probability mass of those slices with category j observations on either side. An exception is the case in which all observations come from a single category, one slice is left unassigned, resulting in a lower probability of $\frac{n-1}{n}$.

The upper probability is equal to the probability mass of of all those slices with category j observations on at least one side. An exception is the case in which c_j in unobserved; in this case the upper probability is equal to $\frac{1}{n}$, as only one slice can be assigned that category.

In the multinomial NPI case, then, the interval in (1.5) is replaced with

$$\left[\max\left(0, \frac{n_j^{\hat{x}_i} - 1}{n^{\hat{x}_i}}\right), \min\left(\frac{n_j^{\hat{x}_i} + 1}{n^{\hat{x}_i}}, 1\right) \right]. \tag{1.6}$$

In this paper trees are generated by the IDM and the multinomial NPI. The splitting criterion is based on an entropy interval comparison as in [8]. For the IDM, algorithms to obtain the minimum and maximum entropy already exist, as in [1] and [4]. For the multinomial NPI, a maximum entropy algorithm is given in [2], and we present a minimum algorithm in section 2. This algorithm will be employed in section 3 to define our splitting criterion. In section 4 the performance of our proposed splitting criterion is evaluated in a simulation study.

2 Minimum and maximum entropy distribution algorithm for multinomial NPI

The maximum entropy algorithm for the multinomial NPI model was already developed and discussed in [2]. Actually two versions to compute the maximum entropy are presented there. One algorithm computes the approximate maximum entropy, which is in structure and proof similar to its IDM counterpart as it assumes the obtained probabilities form a closed and convex set, whereas the other is an exact one, enforcing the restrictions of the probability wheel when assigning probability mass to unobserved categories. In the following only the exact algorithm will be applied.

We now describe an algorithm to calculate the minimum entropy distribution for the f–probability intervals, in the sense of Weichselberger [15]. The intervals for the multinomial NPI were proved to be f–probability intervals in [7].

We begin with a series of lemmas which demonstrate the algorithm's validity, and follow with a schematic outline of the algorithm itself. This algorithm has been adapted from the minimum entropy algorithm for ordinal NPI given in [8].

In what follows $\boldsymbol{L}$ is the vector of lower probabilities and $\boldsymbol{U}$ the vector of the upper probabilities for each category, and we choose elements of $\boldsymbol{L}$ to add mass to until we reach a probability distribution, $\boldsymbol{p}'$. The following four lemmas are required to prove our algorithm minimises entropy. In everything that follows in this section it is assumed that more than one category has been observed; minimising entropy in the case of only one observed category is trivial.

Lemma 1. *Let n_j denote the number of observations of category c_j. For two categories i and j such that n_i and n_j are strictly positive, $U_j - L_j = U_i - L_i = \frac{2}{n}$.*

Proof. Follows directly from the definition of the multinomial NPI model. □

Lemma 2. *Consider elements L_i and L_j, and mass $0 \leq m \leq \frac{2}{n}$. When assigning mass m to either or both of these elements, entropy is minimised by assigning m to c_i if and only if $L_i \geq L_j$, where i and j are interchangeable if $L_i = L_j$.*

Proof. The contributions of p'_i and p'_j to the entropy are $-p'_i \ln(p'_i)$ and $-p'_j \ln(p'_j)$. Note that $H_1(x, y) := -(x \ln(x) + y \ln(y))$ is a concave function in the domain $(x, y) \in [0, 1]^2$. Therefore, for any $0 \leq c \leq m$

$$H_1(p_1 + m - c, p_2 + c) \geq H_1(p_1, p_2 + m),$$
$$H_1(p_1 + m - c, p_2 + c) \geq H_1(p_1 + m, p_2),$$

and hence to minimise H_1, all mass m should be fully assigned to either L_i or L_j. The fact that it should go to the larger of these values also follows from the concave nature of the function. When $L_i = L_j$, the mass must be fully assigned to either, but it makes no difference which is chosen. □

Lemma 3. *The probability distribution $\boldsymbol{p}'$ that minimises entropy is such that $L_i < p'_i < U_i$ holds for at most one i.*

Proof. Assume the contrary, that $L_i + \epsilon_i = p'_i = U_i - \delta_i$ and $L_j + \epsilon_j = p'_j = U_j - \delta_j$ both hold, where all constants in $S := \{\epsilon_i, \epsilon_j, \delta_i, \delta_j\}$ are strictly positive. Further assume $p'_i \leq p'_j$. By the nature of the concave function H_1

$$H_1(p'_i, p'_j) > H_1(p'_i - \min\{S\}, p'_j + \min\{S\})$$

hence minimum entropy has not been achieved. This holds true of any $i \neq j$, meaning at most only one p'_i can have this property. □

Lemma 4. *No mass is assigned to unobserved categories when minimising entropy.*

Proof. By the definition of the multinomial NPI model, $n_i = 0 \Leftrightarrow U_i - L_i = \frac{1}{n}$. We first prove that it is possible to avoid assigning mass to any unobserved category; this follows immediately in the non–trivial case (i.e. $n > 0$) from the definition of the multinomial NPI probability wheel.

It therefore follows that to assign mass to an unobserved category c_k, mass is being "denied" to two observed categories c_i and c_j (again, this follows from the probability wheel). Let $p'_k = m_1 + m_2$, $p'_i = U_i - m_1$, and $p'_j = U_j - m_2$, where $0 < m_1 + m_2 \leq \frac{1}{n} = U_k$. It immediately follows from Lemma 2 that entropy is minimised when $m_1 = 0$ and when $m_2 = 0$. □

Theorem 1. *Entropy is minimised in a structure defined by the multinomial NPI model by assigning the maximum possible mass to the largest element in $\boldsymbol{L}$, then the next largest, and so on until all mass is assigned. When two elements are equally large, choose one of those elements at random.*

Proof. From Lemmas 1 and 4 we will only assign mass to intervals of length $\frac{2}{n}$. Therefore we have that $p'_i \neq L_i \Rightarrow p'_i \in \{U_i - \frac{1}{n}, U_i\}$, where by Lemma 3 $p'_i = U_i - \frac{1}{n}$ holds for at most one i.

If no such i exists, then using Lemma 2 the minimisation algorithm works as follows: assign all $\frac{2m}{n}$ mass (with m an integer) to the m largest elements of L_i, choosing at random between equally large elements.

If one such i, denoted i^*, does exist, we assign $\frac{2m-1}{n}$ mass as above. It is immediately clear that i^* is such that $L_{i^*} = \max_{j \in M}\{L_j\}$ where M is the set of categories with no mass currently assigned to them. All that remains is to demonstrate that the entropy cannot be lowered further by swapping the mass assignment for category c_{i^*} with that of any category $c_k \in M^c$. However, this follows automatically by Lemma 2 for all c_k for which $L_k > L_j$. For any $L_k = L_{i^*}$, swapping as above does not change the entropy. □

Note that this algorithm does not produce the minimum entropy for a general structure. The algorithm can fail when $L_i > L_j > 0$ and $U_j > U_i$ both hold, as it is no longer the case that the stepwise assignment of mass to the largest lower bounds automatically produces the lowest entropy. It might instead be better to assign mass to smaller lower bounds in order to reach larger upper bounds than would otherwise be possible. The NPI multinomial model avoids this problem, as in that model $L_j \geq L_i \Rightarrow U_j \geq U_i$. It is worth noting that the distribution given by this algorithm is not necessarily a unique minimiser. However, the distribution will be unique up to rearranging the elements in ascending order.

Example 1. Consider the case of $K = 5$ classes with six observations $(1, 0, 2, 3, 0)$. From [5] we obtain that the minimum and maximum entropy distribution is contained within the set

$$\frac{1}{6}\left([0,2],[0,1],[1,3],[2,4],[0,1]\right).$$

Applying the exact maximum entropy algorithm as in [2] we obtain the distribution with maximum entropy already in the first step as $\frac{1}{6}(1,1,1,2,1)$.
The minimum entropy algorithm as described above obtains the following *working distributions* in each iteration step:

1. $\frac{1}{6}(0,0,1,2,0)$, 2. $\frac{1}{6}(0,0,1,4,0)$,
3. $\frac{1}{6}(0,0,2,4,0)$.

The entropy interval is then $[0.6365, 1.5607]$. Note that for a distribution over five classes the entropy must lie in the interval $[0, 1.6094]$.

3 Imprecise decision approach to classification trees

We begin by highlighting the differences between the approach in [8] and our approach here. In the former, an imprecise classification tree was defined as a set of classification trees. A decision in each node of the tree was made by comparing the obtained entropy intervals using interval dominance. A tree was then generated for each undominated split variable, hence creating an ensemble of classification trees. Therefore, the work in [8] can be seen as a generalisation of that in [3], which compares only the upper bounds of the entropy intervals, and also allows the generation of multiple trees, though only when considering potential root nodes.

Interval dominance is a strong condition, which means the method in [8] leads in general to a large ensemble of very small trees, as oppose to the smaller ensemble of larger trees created in general by the method in [3]. In particular, this means generating a single tree (and therefore generalising to Abellán and Moral's one–step classification tree algorithm [4]) will in general lead to an overly conservative classification model. In contrast, the Abellán and Moral method can allow splits based on very slight evidence, or even on contradictory evidence which the method ignores. It is not obvious, for example, that a variable with entropy range $[0.39, 0.4]$ should be considered a better choice to split upon than a variable with entropy range $[0, 0.41]$, but the splitting decision in the Abellán and Moral will do so, based just on the difference of 0.01 in the maximum entropy and ignoring the intervals' widths entirely.

Therefore, in this paper we explore whether, when constructing a single tree, there can be found an interval comparison method which is neither so strong as interval dominance, nor so weak as determining the lowest upper bound, and which generates an optimal tree. Our choice to limit consideration to single trees is for the sake of simplicity of comparison; the methods used here can easily be generalised to allow the construction of multiple trees. We refer to the trees generated for this paper as imprecise, as the splitting criterion compares entropy ranges derived from credal sets; note this is a different definition of imprecision to that given in [8]. The split criterion used in this paper is now described.

We note first that any simple comparison of intervals without additional properties is likely to involve one or more of three direct comparisons: comparing the upper bounds, comparing the lower bounds, and comparing the interval lengths. To some extent this third consideration is bound up in the first and second, since of course an interval's length is completely determined by its upper and lower bounds. It is possible that length cannot be completely dealt with by comparison of corresponding bounds, however, otherwise it would be equally easy to choose between intervals $[0.01, 0.95]$ and $[0, 1]$ as to choose between intervals $[0.11, 0.15]$ and $[0.1, 0.2]$, and this is not clearly true. On the other hand, comparing the lengths explicitly would lead to three separate comparisons, which is

arguably overkill, and would require the use of three comparison functions where, for the sake of simplicity, we wish to only use two. We therefore implicitly compare interval length in the comparison of lower bounds shown below. This is done in the comparison of lower bounds rather than that of upper bounds in order to ensure our method is a generalisation of the one found in [2].

Our method of comparing entropy intervals requires two parameters set by the user, that of γ and T_0. We define

$$T = (1-\gamma)A_L + \gamma A_U, \tag{3.1}$$

where A_L and A_U reflect comparisons of the lower and upper bounds respectively (as in Definition 1 below), and $0 \leq \gamma \leq 1$. For each comparison, we choose to split only if $T < T_0$. Therefore the larger the value of T_0 chosen by the user, the less conservative the splitting criterion. Moreover, the greater the value of γ, the more weighting we place upon the comparison of the upper bounds. Therefore $\gamma = 1$ in the Abellán and Moral method, which considers only upper bounds. While in the methods in [8] and [3] the stopping rule is implicitly built-in, in our method we need one explicitly as T is a continuous function of the compared intervals. We now define A_L and A_U.

Definition 1. For the entropy interval $I = [a, b]$ over a data set, and an expected entropy interval $I_i = [a_i, b_i]$ following splitting on attribute variable X_i, we define

$$A_L = \frac{a_i - a}{b_i + |a - a_i|}, \tag{3.2}$$

and further

$$A_U = \frac{\ln(K) - b}{\ln(K) - b_i}. \tag{3.3}$$

Note that A_L is 0 when the lower bounds are equal, and grows larger (smaller) as the lower bound for I_i gets larger (smaller) in comparison to the lower bound for I. Hence a larger value of A_L represents a less desirable split, with respect to the lower bounds. Note also that A_U is equal to 1 when the upper bounds are equal, and gets smaller as the upper bound for I_i gets smaller in comparison to the upper bound for I. Hence a larger value of A_U represents a less desirable split, with respect to the upper bounds. Without any further restriction on when considering upper bound comparison A_U may take values larger than 1 for $b_i > b$, which is covered in what follows.

As noted, in Abellán and Moral's method the splitting is entirely based on the upper bounds comparison. This has the advantage that if there is a split, the maximum entropy is reduced. This property guarantees at least some subgroups which will be more homogeneous. Therefore we also only consider an attribute variable X_i as a split candidate if $b_i < b$.

As T, defined by (3.1), does not satisfy this property of a decreasing expected maximum entropy in the split, we need to enforce more restrictions on our splitting criterion. Therefore we define T^* as follows, dealing with the above mentioned case and interval dominance.

Definition 2. For the entropy interval $I = [a, b]$ over a data set, and an expected entropy interval $I_i = [a_i, b_i]$ following splitting on attribute variable X_i, we define the combined splitting criterion

$$T_i^* = \begin{cases} 1 & \text{if} \quad b_i \geq b \\ T & \text{if} \quad b > b_i \geq a \\ T - 3 & \text{if} \quad a > b_i \end{cases} . \tag{3.4}$$

This ensures that T and therefore A_U is only calculated in situation when $A_U < 1$. Thus in situations when T is actually evaluated it holds that $T \in [-1, 2)$. In the case $a > b_i$ we have I_i interval dominating I. Without the above definition, we would lack the ability to compare among interval dominating split candidates. As $T \in [-1, 2)$ for $b > b_i$ by subtracting three we obtain an always smaller value of T^* for interval dominating split candidates than for those situations where interval dominance does not occur, which allows us to consider both dominated intervals and undominated intervals via the same measure.

The fact that A_L and A_U, along with T, increase as the corresponding bound comparisons become less supportive of a split justifies the choice to split only when $T^* < T_0$. The variable X_{i^*} is chosen to split upon if it is the variable amongst the split candidates with the smallest value of T^*. With T_0 we are able to enforce a specific degree of support for a split. Note that for the Abellán and Moral method, A_L is ignored and A_U is required to be less than one, so the Abellán and Moral method is a special case of our method, with $(\gamma, T_0) = (1, 1)$. A splitting method requiring interval domination may be obtained by setting $T_0 = -1$. With our approach we are able to flexibly adapt the splitting criterion to situations where splits only in case of interval dominance or according to the Abellán and Moral method are favourable.

Although T_0 and γ were said to be chosen by the user in advance, when it is uncertain which actual splitting method to favour, they may be set data-driven, essentially functioning as so called tuning parameters.

4 Simulation

In order to evaluate the performances of the splitting criterion proposed in this paper, simulations were carried out on real–world data sets. The simulation was performed with two major questions in mind: Firstly,

what is the general performance of the proposed splitting criterion and secondly, how does varying the tuning parameters T_0 and γ affect it.

For that purpose 13 different databases from the UCI repository of machine learning [10] were analysed. For each database one classification variable was predicted with the exception of the *Pittsburgh Bridges* database, where five classification variables were independently predicted[1]. Table 1 outlines the number of instances (N), number of continuous and nominal attribute variables (Num and Nom) and total missing values (NA), along with the ranges of the different states of the classification variable (K) and the predicting variables (R).

Database	N	Num	Nom	NA	K	R
abalone	4177	7	1	0	28	3-5
anneal	798	6	32	0	5	1-8
cmc	1473	2	7	0	3	2-5
credit	690	6	9	67	2	2-14
ecoli	336	7	0	0	8	2-5
hepatitis	155	6	13	167	2	2-5
lenses	24	0	4	0	3	2-3
monks1	432	0	6	0	2	2-4
bridges (deck type)	108	1	7	52	2	2-5
bridges (material)	108	1	7	48	3	2-5
bridges (span)	108	1	7	62	3	2-5
bridges (rel. span)	108	1	7	51	3	2-5
bridges (type)	108	1	7	48	7	2-5
po	90	0	8	3	4	2-4
soybean	683	1	34	2337	19	2-5
spect	267	0	22	0	2	2-2
zoo	101	0	16	0	7	2-6

Table 1: Database Overview

In a data pre–processing step any missing values were replaced by the mean or mode for continuous and nominal attributes respectively [2]. Discretisation was applied to the continuous variables by splitting them into five ideally equal frequency intervals, according to the quantiles [3]. Any variables with less than five unique values were not further discretised. Despite being commonly used in such situations, Fayyad and Irani's popular discretisation method [9] was rejected, as for some databases it returned for a notable proportion of predicting variables just one class, essentially removing those variables from the scope of predicting variables. In contrast to previously mentioned decision in the leaves, when there were ties in the most frequent categories, all of those most frequent categories were returned, thus allowing the classification tree to be *imprecise* in the prediction as well. The simulation was completely performed with the open–source statistical programming language R [14].

For each database different configurations of the splitting criterion were analysed: γ was varied over the range $[0, 1]$ and T_0 over $[-1, 1]$. As the configuration $(1, 1)$ corresponds to the maximum frequency criterion of Abellán and Moral, our criterion is implicitly compared to it. Furthermore the case of interval domination is included as T_0 is set to -1 in some configurations. For each setting 50 bootstrap samples were generated and the achieved accuracy on each was reported. On the training data two imprecise classification trees were grown. Both trees employ our proposed splitting criterion, but the underlying models to obtain the set of probability distributions differ: one employs the multinomial NPI and the other a local IDM. The accuracy of the trees was measured in terms of correct classification rate on the determinately predicted instances on the test set [4]. The correct classification rate was evaluated for each tree type on their determinate test data's observations.

To assess the first motivation of the simulation, for each database the optimal configuration of (γ, T_0) is chosen according to the average correct classification rate over the bootstrap sample. However, configuration $(1, 1)$ was not taken into account when evaluating the optimal configuration, because it serves as reference. According to the Wilcoxon signed rank there was a significant difference on a significance level of $\alpha = 0.05$ in the achieved accuracy in favour of our proposed splitting method when comparing it to the Abellán and Moral tress for both the NPI and the IDM approach.

As for the second aspect, there are differences present between the databases, even for the underlying estimation model. For all databases it was found that varying γ resulted in notable variation; only the dataset *po* demonstrated results independent of γ. In general, varying T_0 resulted in very little variation. Overall, the observed behaviour seems reasonable as a change in the weighting may change our splitting criterion drastically, while a change in T_0 only defines the cut point of the splitting criterion when we have non–interval dominating split candidates in a node. Overall, with our method we are not able to advocate a globally optimal γ as it appears database dependent. For the *Pittsburgh bridges (material)* database

[1] This means effectively splitting the database into 5 new databases.

[2] Following the data set description of the *annealing* database, the missing values were considered to be a category in themselves.

[3] Ideally in the sense that no overlapping of categories was permitted and so some categories attained larger/smaller frequencies.

[4] Whenever an observation leads to a prediction of a single class, this observation is said to be determinate, in all other cases, whether two or more classes, it is said to be indeterminate.

low values in γ led to higher accuracy, whereas for *anneal* and *hepatitis* the accuracy was greater for larger values of γ; these comparisons are with respect to the correct classification rate on the IDM–based trees, but similar examples may be found for those based on the NPI method.

Interestingly, there is also a substantial difference between the two tree types: for instance for the *ecoli* data set a high valued γ performs better for the IDM–based trees, but the opposite is true for the NPI–based trees. Moreover on this database for the IDM–based trees the accuracy is higher for $T_0 < 0$ as in comparison to $T_0 > 0$, but for the method based upon NPI the opposite holds.

To further outline the difference between the splitting methods, the performance of each configuration was compared to the one achieved by using the Abellán and Moral splitting. Therefore a Wilcoxon signed rank test was carried out. For most databases there was no significant difference between them for most configurations. However, on the *anneal* and *Pittsburgh Bridges (T or D)* datasets, most configurations achieve a significantly lower accuracy, whereas for the *cmc* and *Pittsburgh Bridges (material)* datasets, with some configurations we are able to significantly improve the accuracy with our splitting criterion.

Furthermore, if there were any significant differences present for a database those were all in the same direction, in the sense that accuracy was either non–increasing or non–decreasing with respect to γ and T_0, with the exception of just three occurrences (two in *soybean* and one in *hepatitis*).

As the previously mentioned difference between the tree–types with respect to changes in γ and T_0 may suggest, substantial differences also exist when comparing variations in those values with the fixed values used in the Abellán and Moral method. However, a significant difference in a certain configuration for the IDM–based tree does not necessarily imply one for the NPI–based and vice versa. On the other hand, for most databases, if there are significant differences present, they are in the same direction, i.e both are greater/less. Exceptions are the *spect* and *zoo* where on some configurations the accuracy is significantly improved using the IDM, but for the NPI on some (other) configurations we are predicting significantly worse.

In general, taking all databases into account, there is only a small difference between our splitting criterion and the Abellán and Moral one. On some databases we are able to improve the achieved accuracy with a certain database specific configuration of γ and T_0, while on others we are losing some accuracy for some settings. However, in most cases there is a significant difference between the Abellán and Moral splitting approach and our more general (and also more complicated) approach. The choice of the underlying probability model naturally influences the obtained results. Our results concur with [2] in that we find no significant difference between the NPI– and IDM–based trees when comparing them according to their best performance on each database. However, the NPI approach has a slightly poorer performance with our method in comparison to the Abellán and Moral splitting criterion. Generally, we are not able to identify an overall optimal configuration of (γ, T_0). This difficulty in predicting the effects of a change in parameter casts doubt on the ability of users to sensibly choose parameter values for the current model.

In our simulation we did not consider a comparison of our method to the underlying ID3 splitting mechanism. As [4] pointed out in their simulations, their splitting method has the ability to successfully compete against the even more advanced splitting algorithm C4.5.

5 Conclusions and further aspects

In this paper an approach to building classification trees using entropy range comparisons was outlined and tested. This process required the creation of an entropy minimisation algorithm, which was presented here for the multinomial NPI method. This algorithm was then used to compare trees built using the splitting criterion suggested in [4], which considers only the upper bounds of the entropy interval, and our method, which compares both upper and lower bounds of the entropy interval, with a user–defined weighting on these two comparisons determining which is the more important. A second user–defined criterion determines the amount of dissimilarity between entropy intervals necessary to justify a split; the ranges of these two user–defined criteria means our model includes both that described in [2] (which applies the model in [4] to the NPI case) and that described in [8] (in which interval dominance is required to allow splitting). These methods were compared over 13 datasets, and the resulting simulation bore interesting results. Whilst it is not the case that there exists a specific combination of user–defined criteria that improves upon consideration of the upper bounds alone, it is possible in many cases to find a combination that does improve upon that method for the specific dataset. Moreover, our results support the hypothesis that in situations in which comparison of upper bounds strongly support splitting it can make a noticeable difference to accuracy whether or not splits are allowed for variables with associated in-

tervals which have higher lower bounds than the interval for the dataset.

Therefore it can be stated that our method has the potential to improve accuracy, but more work is required in determining under what circumstances this is the case. Related to this, further work is required in justifying this method or one similar to it through a decision theoretical foundation.

It also remains to be explored how our method performs in comparison with [3] when the former is used to generate ensemble trees. As mentioned in the text, reducing to the case of a single tree allows for quicker and more easily interpreted comparisons, but our method was created with ensemble trees in mind, and this should be considered further.

To allow for a comparison with precise classifiers future simulations will also include a precise classifier. Furthermore an investigation about the tree's actual length for the optimal configuration is worth carrying out. Larger trees, especially with ensembles in mind, induce a higher computational cost, even if it decreases in the future with more powerful hardware architecture.

Appendix

Algorithm 1 gives an outline of the minimum entropy algorithm as proposed in section 2.

When considering its computational complexity, it mainly depends on the ordering of the $[l_i, u_i]_1^n$. The proposed algorithm requires generally the least steps, when $[l_i, u_i]_1^n$ is sorted according to decreasing l_i. Any of the popular sorting algorithms may be applied to obtain such a sorting, with complexity ranging from $O(n)$ to $O(n^2)$. The initialisation step means just copying l to p and generation of and index set. Due to the special ordering of l, $getMaxIndex$ in the **while** () **do**-loop finds the return value immediately as it is j when in the jth loop. Furthermore, because of the special representation of the multinomial NPI on a probability wheel, it is immediately clear that the **while** () **do**-loop has at maximum $\lceil \frac{n}{2} \rceil$ iterations. Therefore the algorithm without the sorting runs in linear time. Hence the computational most time intensive part is the chosen sorting algorithm.[5].

In the following the splitting procedure is outlined, considering the splitting process within a node N. Let $\mathcal{L}_N$ be the set of the attribute variables which are not used splitting variables on the path from the root node to N. Finding the optimal split requires three steps:

Algorithm 1 Minimum Entropy Algorithm for NPI

Input:	F-probability intervals $[l_i, u_i]_1^n$ as generated by the NPI
Output:	A probability distribution $\hat{p} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n)$

Helping functions:

Sum(x):	returns the sum of the elements of array x
getMaxIndex(x, S):	returns the first index of the maximum value of the array x considering only indices in S

Initialization: S $\leftarrow 1, \ldots, n$

```
minEntropyNPI(l, u, p̂){
  for (i = 1 to n) do {p̂_i ← l_i}
  mass ← 1 − Sum(p̂)
  while (mass > 0) do {
    index ← getMaxIndex(p̂, S)
    d ← u_index − p̂_index
    if (d ≤ mass) then {
      p̂_index ← u_index
      S ← S − {index}
      mass ← mass − d
    } else {
      p̂_index ← p̂_index + mass
      mass ← 0
    }
  }
}
```

1. T_i^* is calculated for each $X_i \in \mathcal{L}_N$[6];
2. X_{i^*} is chosen as reasonable splitting candidate among the X_i in $\mathcal{L}_N$, where $T_{i^*}^* = \min_i (T_i^*)$;
3. A comparison of $T_{i^*}^*$ and T_0 is made. Only if $T_{i^*}^* < T_0$ is X_{i^*} chosen as the split variable, otherwise the node N is declared terminal.

Acknowledgement

We are grateful for the remarks of three anonymous reviewers and feel that some remarks need to be separately considered in further research. The work of R. Crossman has been partly supported by the Spanish "Consejería de Economía, Innovación y Ciencia de la Junta de Andalucía" under project TIC-6016.

References

[1] Joaquín Abellán. Uncertainty measures on probability intervals from the imprecise Dirichlet

[5] In the simulation the Shell-Sort algorithm was applied as it is implemented in R [14]

[6] The entropy interval I_i required to calculate T_i^* is obtained in the same way as in [8]

model. *International Journal of General Systems*, 35(5):509–528, 2006.

[2] Joaquín Abellán, Rebecca M. Baker, and Frank P.A. Coolen. Maximising entropy on the nonparametric predictive inference model for multinomial data. *European Journal of Operational Research*, 212(1):112–122, 2011.

[3] Joaquín Abellán and Andres Masegosa. An ensemble method of using credal decision trees. *European Journal of Operations Research*, 205(1):218–226, 2010.

[4] Joaquín Abellán and Serafín Moral. Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12):1215–1225, 2003.

[5] Rebecca M. Baker. *Multinomial Nonparametric Predictive Inference: Selection, Classification and Subcategory Data*. PhD thesis, 2010. www.etheses.dur.ac.uk/257/.

[6] F.P.A. Coolen and T. Augustin. Learning from multinomial data: a nonparametric alternative to the imprecise dirichlet model. In *ISIPTA'05: Proceedings of the Fourth International Symposium on Imprecise Probability: Theories and Applications*, pages 125–134, Carnegie Mellon, 2005. SIPTA.

[7] Frank P. A. Coolen and Thomas 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):217–230, 2009.

[8] Richard J. Crossman, Joaquín Abellán, Thomas Augustin, and Frank P. A. Coolen. Building imprecise classification trees with entropy ranges. In F. Coolen, G. de Cooman, Th. Fetz, and M. Oberguggenberger, editors, *ISIPTA'11: Proceedings of the Seventh International Symposium on Imprecise Probability: Theories and Applications*, pages 129–138, Innsbruck, 2011. SIPTA.

[9] Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Thirteenth International Joint Conference on Articial Intelligence*, volume 2, pages 1022–1027. Morgan Kaufmann Publishers, 1993.

[10] Andrew Frank and Arthur Asuncion. UCI machine learning repository, 2010.

[11] B.M. Hill. Posterior distribution of percentile: Baye's theorem for sampling from a population. *Journal of the American Statistical Association*, 63:677–691, 1968.

[12] John R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986.

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

[14] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[15] Kurt Weichselberger. The theory of interval–probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, 24(2–3):149–170, 2000.