MASTER THESIS

by

Paul Fink

Ensemble methods for classification trees under imprecise probabilities

---

Supervision:
Prof. Dr. Thomas Augustin

Department of Statistics
Ludwig–Maximilians–University
Munich
May 29, 2012

Abstract

In this master thesis some properties of bags of imprecise classification trees, as introduced in Abellán and Masegosa (2010), are analysed.

In the beginning the statistical background of imprecise classification trees is outlined – starting with an overview on measuring uncertainty within the concept of Dempster–Shafer theory is presented, followed by a discussion of its application in a tree–growing–algorithm, which employs the so–called Imprecise Dirichlet Model in the splitting process.

The motivation of so–called ensemble methods is to reduce the instability of a single classification tree, increasing its predictive accuracy, but at cost of structural interpretability. A description of the well known ensemble methods, such as bagging, random forests and boosting, is given along with two approaches, generating the ensemble by allowing more than one splitting variable within a node.

In the next step a bag of imprecise classification trees is generated; following, its sensitivity in relation to

- different ensemble aggregation/fusion rules (majority voting, disjunction and average rule),
- the external stopping criterion of minimal observations within a node and
- the main parameter of the Imprecise Dirichlet Model

is analysed by a simulation study.

The results of the simulation indicate that the commonly applied majority voting rule is also a fair choice for imprecise classification ensembles. Regarding the external stopping criterion the simulation indicates that such may be neglected, while the parameter does highly affect the predictive accuracy, favouring lower values of it.

. . . . . .

Contents

Chapter 1

Introduction

The task of classifying items according to some feature variables may be accomplished by a broad methodology. Generally a classifier is a mapping of feature variables to a certain state of classification variable. A classifier may be used to understand the underlying structure on how the classification variable is related to the features. Another application is to predict the class of new items on the basis previous items, for which the true class is known to the researcher. No matter the intention classification is a statistical inference as a structure is inferred on some known observations, referred to as *learning data/sample*.
In this master thesis the focus is on the special case of classification trees, which perform a recursive partitioning of the sample space. As demonstrated in Breiman et al. (1984)[1] those classification trees are sensitive to noise in the data. In the classifying context Zaffalon (2002) proposed the *Naive Credal Classifier* (NCC) as a generalization of the *Naive Bayes Classifier* (NBC). It is an approach accounting for the general imprecision in the learning sample. The NCC is allowed to output a set of states of the classification variable, instead of being restricted to singletons. In Zaffalon et al. (2003) they demonstrate the superiority of the NCC over the NBC in a noisy context, due to missing values. The concept of credal classification is also introduced by Abellán and Moral (2003*a*) but with a different motivation. They derive their credal classifier by introducing an imprecise splitting criterion into Quinlan's ID3 algorithm. The splitting criterion is based on measurements of uncertainty in Dempster-Shafer theory.
Breiman's approaches to tackle the tree's instability to noisy data were bagging (Breiman (1996)) and its generalization random forests (Breiman (2001)). Both techniques belong to the category of *ensemble methods*. In an ensemble method a *base learner* is applied multiple times to a varying learning sample. The output of the ensemble is an aggregate/fusion over the base learners' output in the ensemble itself. Another popular ensemble technique is Boosting (Freund and Schapire (1966)).
Recently Abellán and Masegosa (2010) studied the performance of bagging their imprecise classification trees in comparison to bagging precise trees.
In the following the focus is set on the comparison of different aggregating approaches for a bag of imprecise classification trees.

[1] Breiman et al. (1984), p. 156ff

In chapter 2 the basic concept of a classification tree is outlined. After it follows a discussion on measure of uncertainty in the context of Dempster-Shafer theory (chapter 3). For better insight Walley's Imprecise Dirichlet Model is described in chapter 4, as it is heavily involved in the actual growing of an imprecise classification tree as described in chapter 5. In the then following chapter 6 common ensemble methods - bagging, random forests and boosting as well as TWIX ensembles - are sketched, finishing it with a discussion on the application of frequently used fusion rules on imprecise bags. Chapter 7 states the results of a simulation regarding the impact of the fusion rules on the predictive accuracy. Finally, concluding remarks and an outlook on further research on the topic are given in chapter 8.

Chapter 2

Classification trees

2.1 Introduction

The purpose of classification trees is to classify an observation according to feature/attribute variables. It is achieved by recursively partitioning the data space into rectangular disjoint subspaces. Each area is then assigned to just one value of the classification variable.
In order to build the tree for some of the observations the values of the classification variable need to be known, on which the tree will then be constructed. Thus the method of classification trees is a so–called supervised learning technique. In the following, restrictions to the data and a general tree building procedure will be presented.

2.2 Restriction to the data

It is obvious that the classification variable C needs to be categorical. Although an ordinal nature of it may be acknowledged, it will not be taken into account in this work. As for the feature variables (X_1 to X_n) there are generally no restrictions on the scale: they may either be continuous or categorical (ordered and unordered). The most popular classification tree algorithms, Breiman et al. (1984) CART algorithm produces binary splits for both scales, whereas Quinlan (1986) C4.5 produces binary splits for continuous predictor variables and as many nodes as categories of the prediction variable for categorical variables (Often referred to as *k–array–splitting*). In here a k–array splitting algorithm of Abellán and Moral (2003*a*) will be used. To split a continuous feature variable in the context of k–array–splitting, two approaches are worth noting here. A naive one would force the continuous variable on a nominal scale, thus enforcing k–array–splitting. However for those variables with many different values, especially not exhaustive in the training sample, a rather suboptimal tree is grown.[1] In a more elaborate approach, as in C4.5, the continuous variable is binarily split. For each value of such a feature variable the split criterion is calculated, assuming a binary split, i.e. all values smaller than the cutpoint are assigned

[1] A summary on how the number of categories influences the splitting itself is outlined in chapter 5.1.

to the left side, all others to the right. In this master thesis however all feature variables are assumed to be categorical.

2.3 Tree building

In terms of the above section, we have a data set $\mathscr{D}$ with a set $\mathscr{L}$ of predictor variables $\{X_i\}_n^1$, of any scale. Furthermore $\mathscr{D}$ contains the discrete classification variable C with states in in $\Omega_C = \{c^1, c^2, \ldots, c^{|\Omega_C|}\}$.
The tree is grown from its root, the complete data, to its leaves, disjoint subsets, in a recursively applied splitting procedure. In each splitting step an optimal variable is chosen according to this, the data in the node are optimally assigned to daughter nodes. It appears that the splitting in a node can be described as a twofold optimization problem.

Step 1 Finding an optimal split point for each of predictor variables each in terms of a pre–chosen impurity criterion.

Step 2 Splitting the data in the node according to the predictor variable which produces the greatest decrease of the impurity criterion at its optimal split point, calculated in Step 1.

Assuming the actual node is $\mathbf{C}$ and $\{\mathbf{C}_{t_i}\}$ is the set of daughter nodes produced by splitting X_i at point t_i. In case of binary splitting $\{\mathbf{C}_{t_i}\}$ is reduced to $\{\mathbf{C}_{\text{left}\,t_i}, \mathbf{C}_{\text{right}\,t_i}\}$.
In the first step the reduction in impurity criterion $\mathbf{\Delta IC}$ is achieved by comparing $\mathbf{IC}(\mathbf{C})$ to $\mathbf{IC}(\{\mathbf{C}_{t_i}\})$. Its actual calculation is done by the plug–in estimator $\widehat{\mathbf{\Delta IC}}$. The optimal split point is then calculated by:

$$t_i^* = \arg\max_{t_i} \widehat{\mathbf{\Delta IC}}(\mathbf{C}, \{\mathbf{C}_{t_i}\}) \qquad \forall i = 1, \ldots, n. \tag{2.1}$$

In case of k–array–splitting or a binary variable X_i no optimization is needed, as there is only one possible split point for the variable so Step 2 is omitted.
The second step reveals a less feasible optimization. In order to find the optimal splitting variable X_{i^*} one has to solve

$$i^* = \arg\max_{i} \widehat{\mathbf{\Delta IC}}(\mathbf{C}, \{\mathbf{C}_{t_i^*}\}) \tag{2.2}$$

with t_i^* being the optimal split for each variable as calculated in equation (2.1). The knot $\mathbf{C}$ will then be split at the variable X_{i^*} at its split point t_i^*, resulting in the set of daughter nodes $\{\mathbf{C}_{t_i^*}\}$. For each of these daughter nodes the algorithm is applied again.
This naive approach leads obviously to *pure* leaves, i.e. all observations in it have the same state of the classification variable. In fact it leads to an extremely over–fitting of the learning data. Hence some restrictions need to be incorporated both in the actual tree growing and as a post–processing step after the tree is completely built.
In the first case the splitting of a node into daughter nodes is restricted to a pre–chosen minimal number of observations, passed to each of the successor nodes. Another restriction limits the tree to a user–specified depth.
The post–processing is called *pruning*. It takes into account the generalization aspect of the tree: By moving from the leaves to the root it cuts back branches which lead to an increase in the generalization error.

Chapter 3

Measuring uncertainty in Dempster–Shafer theory

The underlying concept of Dempster–Shafer theory[1] is a general mass assignment. Provided a finite set X of size $|X| = n$ and its power set $\mathscr{P}(X)$ a *basic probability assignment* is a mapping function m of

$$m : \mathscr{P}(X) \to [0, 1]$$

with the restrictions

$$m(\emptyset) = 0 \qquad \text{and} \qquad \sum_{A \subseteq X} m(A) = 1 \; .$$

Contrary to point–probability–assignments, $m(A)$ gives a degree of belief for an element $x \in X$ being in any subset of A, but not favouring one particularly. Applying the above notation, one may obtain the *Belief* Bel and the *Plausibility* Pl as functions on the assignment.

$$\begin{aligned} Bel(A) &= \sum_{B \subseteq A} m(B) \\ Pl(A) &= \sum_{B \cap A \neq \emptyset} m(B) \end{aligned}$$

which are linked on the condition of

$$Pl(A) = 1 - Bel(\bar{A}).$$

According to the theory of Yager total uncertainty could be divided into *randomness* and *non–specificity*.[2] While randomness measures how information[3] is split on disjoint subsets, i.e. it attains its maximum when the information is split uniformly over the disjoint subsets, non–specificity takes the imprecision of information into account, i.e. has higher values for an assignment m for a set with more than one element.

[1] cf. Dempster (1967) and Shafer (1976)
[2] Yager (1983), p. 252, 259
[3] induced by the mass assignment m

In Harmanec and Klir (1994) a chronological overview of proposed functions to measure total uncertainty is given. They developed a measure of total uncertainty generalizing the Shannon–Entropy. It is motivated as any body of evidence on $\mathscr{P}(X)$ may be seen as a set of constraints defining probability functions on X acceptable. Among those acceptable functions there is one which maximizes the Shannon–Entropy. According to them the attained value of the maximal Shannon–Entropy is a reasonable measure of the total uncertainty as it furthermore satisfies the required properties of a measure for total uncertainty in Dempster–Shafer theory.[4]

Definition 1. *Harmanec and Klir's measure of Total Uncertainty*[5]
Let X, Bel denote a frame of discernment and a belief function on $\mathscr{P}(X)$, respectively, and let $\langle p_x \,|\, x \in X\rangle$ denote a probability distribution on X. Then we define the amount of uncertainty contained in Bel, denoted as $AU(Bel)$, by

$$AU(Bel) = \max\left\{ -\sum_{x\in X} p_x \log_2 p_x \right\}, \tag{3.1}$$

where the maximum is taken over all distributions $\langle p_x \,|\, x \in X\rangle$ that satisfy the constraints:

(a) $p_x \in [0,1]$ for all $x \in X$ and $\sum_{x\in X} p_x = 1$;

(b) $Bel(A) \leq \sum_{x\in A} p_x \leq 1 - Bel(\bar{A})$ for all $A \subseteq X$.

The above measure is then employed by Abellán and Moral (2000) to measure the randomness in a general credal set. They call it the *Upper Entropy* $G(m)$ where m is the mass assignment.
In order to measure the non–specificity they generalize Dubois and Prade (1985) measure of non–specificity. It is defined, using the Möbius Inverse function. For a deeper understanding at first there are three preliminary definitions given. All are derived from their basic definitions in the context of Dempster–Shafer theory.

Definition 2. *Lower Probability function on Credal Set*[6]
Let C be a c.s.p.d.[7] *on a universal X. We define the following capacity function:*

$$f(A) = \inf_{P\in C} P(A), \qquad \forall A \in \wp(X)$$

where $\wp(X)$ is the power set of X.

Definition 3. *Möbius Inverse*[8]
For any mapping $f_{\mathscr{P}} : \wp(X) \to \mathbb{R}$ [...] the mapping $m_{\mathscr{P}} : \wp(X) \to \mathbb{R}$, given by

$$m(A) = \sum_{B\subseteq A} (-1)^{|A-B|} f(B), \qquad \forall A \in \wp(X),$$

will be called Möbius Inverse of f.

[4] The properties are given in Harmanec and Klir (1994), p. 408f
[5] Harmanec and Klir (1994), p. 412f
[6] Abellán and Moral (2000), p. 360f
[7] convex set of probability distributions; note from the author
[8] Abellán and Moral (2000), p. 361

Definition 4. *Focal elements*[9]
Let C be a c.s.p.d. on a universal X, f its minimum lower probability as in Definition 2 and let m be its Möbius Inverse. We say that m is the assignment of masses *of C: And we call any $A \in X$ such that $m(A) \neq 0$, a focal element of m.*

In this context $m(\emptyset) = 0$. Furthermore the sum over all elements in the power set of X equals 1: $\sum_{A \in \wp(X)} m(A) = 1$. With definition 4 the index set of the sum is reducible to the focal elements.
With the above definitions a generalization of Dubois and Prade non–specificity measure U[10] may be obtained by definition 5.

Definition 5. *Non–specificity measure*[11]
Let C be a c.s.p.d. on a universal X. Let m [be; note from the author] the assignment of masses associated to C. We define [$IG(C)$; note from the author] the non-specificity of C as

$$IG(C) = \sum_{A \subseteq X} m(A) \ln(|A|). \tag{3.2}$$

Maeda and Ichihashi (1993) added to the entropy like measure (3.1) the non–specificity measure (3.2) to obtain a measure of total uncertainty for any basic probability assignment.[12] Accordingly, Abellán and Moral (1999) applied this measure to the case of credal sets:

$$UT(m) = I(m) + G(m)\ ,^{13} \tag{3.3}$$

where m is the mass–assignment.
In an addition to this uncertainty measure, Abellán and Moral (1999) proposed a factor to correct the uncertainty measure. They call it the *Kullback–Factor*. It takes the distance of the uniform distribution to the frontier set of the convex set of probabilities associated with m into account, in case it is included in it.[14]
However in their following work they changed their opinion on a reasonable measure of total–uncertainty for credal sets. As $G(m)$ also increases subject to non–specificity they argue that adding $I(m)$ "*gives rise to overweight imprecision*",[15] settling for the randomness measure $G(m)$ finally.
While Harmanec and Klir do not justify their measure intrinsically, Abellán and Moral (2005) state a plausible explanation within the theory of evidence: The probability distribution with maximum entropy is the one with minimum payment under a logarithmic scoring rule. Hence the less certain one is about the true value, the higher will be the value of the upper entropy and vice versa. Abellán and Moral point out that this does not satisfy a substitution of the whole credal set with the distribution of maximum entropy, but may be considered as a reasonable measure of uncertainty within it.[16]

[9] Abellán and Moral (2000), p. 361
[10] Dubois and Prade (1985), p. 282: Equation 16
[11] Abellán and Moral (2000), p. 361
[12] Maeda and Ichihashi (1993), p. 387 equation 11)
[13] With I being another notation of IG (equation (3.2)) and G of AU (equation (3.1))
[14] A description and the properties are given in Abellán and Moral (1999)
[15] Abellán and Moral (2005), p. 239
[16] Abellán and Moral (2003*c*), p. 5

In order to measure the uncertainty within a credal set, regardless the measure, the credal set itself needs to be defined and estimated when dealing with actual data. In order to estimate such credal sets, Abellán and Moral (2005) employ the Imprecise Dirichlet Model locally, which was introduced in Walley (1996). A justification as reasonable choice to estimate a credal set was proposed in Zaffalon (2002) to obtain a credal classifier, assuming that the classification variable's values are drawn from a multinomial distribution.
For a better understanding of the tree-building process a short introduction to Walley's Imprecise Dirichlet Model is given.

Chapter 4

Imprecise Dirichlet Model[1]

In the context of Bayes multinomial sampling a Dirichlet prior distribution is a common choice as it is conjugated to the multinomial distribution of the underlying observed data.
The assumptions of the standard multinomial model are as follows:
Considering a sample space $\Omega = \{\omega_1, \ldots, \omega_K\}$ with $K \geq 2$ mutually exclusive categories, N observations have been sampled independently and uniformly according to the probability distribution $\mathbb{P}(\omega_j) = \pi_j$ with $j = 1, \ldots, K$ and restrictions to $\pi_j \geq 0$ and $\sum_{j=1}^{K} \pi_j = 1$. The number of the occurrences is recorded for each category, denoted by n_j for category ω_j. Obviously the category occurrences are non–negative integers and sum up to N: $\sum_{j=1}^{K} n_j = N$. To simplify the notation the category occurrences $n = (n_1, \ldots, n_K)$ and its probabilities $\pi = (\pi_1, \ldots, \pi_K)$ are vectorized.
Under those assumptions the random numbers for n_j follow a multinomial distribution and the observed likelihood function of n is

$$L(n|\pi) \propto \prod_{j=1}^{K} \pi_j^{n_j}.$$

As stated above a Dirichlet prior is a common choice for π: $\pi \sim Dir(\alpha_1, \ldots, \alpha_K)$. Its probability density function is then:

$$\mathrm{f}(\pi) \propto \prod_{j=1}^{K} \pi_j^{\alpha_j - 1}.$$

However, in its motivation of the Imprecise Dirichlet Model, Walley gives a different parametrisation of it. Each of the α_j is divided into $\alpha_j = s \cdot t_j$ where $s > 0$ characterises the distribution and $0 < t_j < 1, \sum_{j=1}^{K} t_j = 1$, here $\pi \sim Dir(s, t)$. In this version t_j is the expectation for category j: $\mathbb{E}(\pi_j) = t_j$.
Using the latter notation the probability density function becomes:

$$\mathrm{f}(\pi) \propto \prod_{j=1}^{K} \pi_j^{st_j - 1}.$$

[1] The whole chapter is based on Walley (1996)

The posterior distribution is obtained by multiplying the prior of π and the observed likelihood function of n, resulting in

$$L(\pi|n) \propto \prod_{j=1}^{K} \pi_j^{st_j+n_j-1},$$

which is in the form of a probability density function of a $Dir(N+s, t^*)$ distribution with $t^* = (st_j + n_j)/(N+s)$.
In case of a standard multinomial model the parameters α, t and s respectively, are fixed in the sense that the model takes just one value (for the vectors: one vector of values) of it into account. Thus it leads to precise results when calculating the expected posterior mean for instance.
Walley lifts the restriction to precise results by taking not a single prior distribution $Dir(s,t)$ but rather all possible $Dir(s,t)$ distributions, short $Dir(s)$, for a fixed s where t satisfies the aforementioned properties. The so constructed set is called $\mathcal{M}_0$. Taking the whole set as prior means that no value specific assumptions concerning the chances π are made, thus modelling near prior ignorance. However symmetry of the categories is still modelled.
Taking a set as prior naturally leads to a set $\mathcal{M}_N$ of posterior distributions $Dir(N+s, t^*)$. Note that $N+s$ is a fixed value too. As the posterior distribution is a set rather than a single function there is not a point probability but a set of probabilities for a specific data situation. The lower and upper bounds, posterior lower and upper probabilities, are calculated by the probability infimum and supremum of an event of all Dirichlet distributions in $\mathcal{M}_N$.[2]
So the probability that n_j occurrences of category j appear in N tries is equal to the posterior mean of π_j. As aforementioned it is the set of all $t_j^* = (st_j + n_j)/(N+s)$ which belong to Dirichlet distributions in $\mathcal{M}_N$. The posterior lower probability is then obtained by minimizing t_j^* with respect to t_j:

$$\begin{aligned}
\underline{\mathbb{E}}(\pi_j|n) &= \liminf_{t_j} t_j^* \\
&= \liminf_{t_j} \frac{st_j + n_j}{N+s} \\
&= \lim_{t_j \to 0} \frac{st_j + n_j}{N+s} \\
&= \frac{n_j}{N+s}\,. \qquad (4.1)
\end{aligned}$$

Analogously the corresponding posterior upper probability is calculated:

$$\begin{aligned}
\overline{\mathbb{E}}(\pi_j|n) &= \limsup_{t_j} t_j^* \\
&= \limsup_{t_j} \frac{st_j + n_j}{N+s} \\
&= \lim_{t_j \to 1} \frac{st_j + n_j}{N+s} \\
&= \frac{s+n_j}{N+s}\,. \qquad (4.2)
\end{aligned}$$

[2]For a theoretical justification see Walley (1991)

Those two equations will be needed later in the tree generation process.
Although the bounds of the interval depend on the actual occurrences n_j, the width is independent of the category and only based on the value of s and the number of samples N:

$$\overline{\mathbb{E}}\left(\pi_j|n\right) - \underline{\mathbb{E}}\left(\pi_j|n\right) = \frac{s+n_j}{N+s} - \frac{n_j}{N+s} = \frac{s}{N+s}\ . \tag{4.3}$$

As s increases, so does the interval width and thus the imprecision generated by the IDM. However for a fixed s the more information is available to the model, i.e. increasing number of N, the smaller the interval gets, reflecting the information gain.
Another notably feature of the IDM, which will be used later on, is the possibility to easily combine categories, for instance categories i_1 and i_2. Then $\pi_i = \pi_{i_1} + \pi_{i_2}$ directly from the properties of the Dirichlet distribution. This leads to a generalisation of (4.1) and (4.2):

$$\mathbb{E}\left(C|n\right) = \sum_{\omega_j \in C} t_j^* = \frac{st(C)+n(C)}{N+s}\ ,$$

where $t(C) = \sum_{\omega_j \in C} t_j$ and $n(C) = \sum_{\omega_j \in C} n_j$ is the number of occurrences.
Following the same rationale as in (4.1) and (4.2) the lower and upper bounds are obtained to

$$\begin{aligned}
\underline{\mathbb{E}}\left(C|n\right) &= \liminf_{t(C)} \sum_{\omega_j \in C} t_j^* \\
&= \liminf_{t(C)} \frac{st(C)+n(C)}{N+s} \\
&= \lim_{t(C)\to 0} \frac{st(C)+n(C)}{N+s} \\
&= \frac{n(C)}{N+s}
\end{aligned} \tag{4.4}$$

and

$$\begin{aligned}
\overline{\mathbb{E}}\left(C|n\right) &= \limsup_{t(C)} \sum_{\omega_j \in C} t_j^* \\
&= \limsup_{t(C)} \frac{st(C)+n(C)}{N+s} \\
&= \lim_{t(C)\to 1} \frac{st(C)+n(C)}{N+s} \\
&= \frac{s+n(C)}{N+s}.
\end{aligned} \tag{4.5}$$

Walley gives a reasonable interpretation of the parameter s as the number of hidden instances and N the already revealed one, thus interpreting (4.4) and (4.5) as relative frequencies of the event C.[3]

[3] cf. Walley (1996), p. 10

For the parameter s Walley does not give any clear preference of a value, yet he slightly advocates values of $1 \leq s \leq 2$. As demonstrated in (4.3) smaller values of s produce more precise results, i.e. smaller intervals; deciding on greater values may lead to overcautious results.[4]
Armed with a variety of different imprecision measurements and a model to estimate the relative frequencies of a category state, we can move on to the process of growing a classification tree using imprecise probabilities.

[4]The choice of $s = 1$ or $s = 2$ are employed in e.g. Walley (1996), Abellán and Moral (2003*a*) and Abellán and Moral (2005); Bernard (2005) associates different values of s with certain types of prior distributions in case of a Dirichlet Model (slide 19)

Chapter 5

Classification tree under imprecise probabilities

In the previous chapters the basic tools of growing a classification tree from the root to its leaves were given. The algorithm employed by Abellán and Moral (2003*a*) is based on Quinlan's C4.5 algorithm.
As the notation in Abellán and Moral is intuitive, it will also be deployed in here.
In here the predictor variables are limited to the nominal or ordinal scale, however the additional information of the order in an ordinal variable is not taken into account and hence treated as nominal. The predictor variables in $\mathscr{L}$[1] are discrete and take values/states in $\Omega_{X_i} = \{x_i^1, x_i^2, \ldots, x_i^{|\Omega_{X_i}|}\}$. As stated in chapter 2.3 the classification variable is also of a nominal scale, indexed by c_j for $j = 1, 2, \ldots, |C| = k$.
To retain the path from the root node to any other Abellán and Moral introduce the *configuration* σ.

Definition 6. *Configuration*[2]
Let $\{X_i\}_n^1$ *be a set of discrete variables with values in the finite sets* Ω_{X_i}*, respectively. We call a configuration* σ *of* $\{X_i\}_n^1$ *any m-tuple*

$$\left(X_{r_1} = x_{r_1}^{t_{r_1}}, X_{r_2} = x_{r_2}^{t_{r_2}}, \ldots, X_{r_m} = x_{r_m}^{t_{r_m}}\right) \tag{5.1}$$

where, $x_{r_j}^{t_{r_j}} \in \Omega_{X_{r_j}}$*,* $j \in 1, \ldots, m$*,* $r_j \in 1, \ldots, n$ *and* $r_j \neq r_h$ *with* $j \neq h$*. A configuration* σ *is thus an assignment of values for some of the variables in* $\{X_i\}_n^1$*.*

Applying the above notation, X^σ is the set of observations compatible with the configuration σ. A configuration allows to identify such observations satisfying the conditions induced by the configuration.
As seen in chapter 3 Abellán and Moral favour the Upper Entropy as impurity measure. In the context of classification trees it serves as splitting criterion in a node, defined by

[1] See chapter 2.3
[2] Abellán and Moral (2003*a*), p. 1218

Definition 7. *Upper Entropy on Credal Set*
Given a credal set $\mathscr{P}$ on a variable U, then the upper entropy is given by

$$G(\mathscr{P}) = \max_{P \in \mathscr{P}} \left\{ -\sum_{x \in U} P(x) \ln(P(x)) \right\} . \tag{5.2}$$

This follows immediately from equation (3.1). The probability distribution with maximal entropy is called Upper Entropy Distribution. The Upper Entropy Distribution is the probability distribution in the credal which minimizes the distance to the uniform distribution. It may be interpreted as the most uninformative distribution in the credal set and hence the most cautious one in estimating the probabilities of the different states.

5.1 Splitting in a node

In this section the Imprecise Dirichlet Model will be applied to generate estimators of the class–probabilities within a node. Based on those the distribution with maximum entropy is calculated and deployed as plug–in estimator to the impurity measure. The advantage of an imprecise model over a precise one is an increase in prediction robustness.
Starting in the root node (or any subsequent node) the Imprecise Dirichlet Model is locally applied to the node's configuration to obtain the probability for each state of the classification variable. Locally means here that the model is based exclusively on those observations complying with the node's configuration. However the characterizing parameter s is chosen globally. As the value of s influences both the size of the grown tree and its accuracy, which will be shown later, the more conservative approach of Walley (1996) and Abellán and Moral (2003*a*) for $s = 1$ or $s = 2$ is reasonable.
As presented in chapter 4 the posterior lower and upper probabilities of a state $c_j \in C$, given a configuration σ

$$\left[\underline{P}^{\sigma}_{c_j}, \overline{P}^{\sigma}_{c_j}\right] = \left[\underline{P}\left(C = c_j | X^{\sigma}\right), \overline{P}\left(C = c_j | X^{\sigma}\right)\right] = \left[\frac{n^{\sigma}_{c_j}}{N^{\sigma} + s}, \frac{n^{\sigma}_{c_j} + s}{N^{\sigma} + s}\right], \tag{5.3}$$

where $n^{\sigma}_{c_j}$ is the number of observations in configuration σ in state c_j and $N^{\sigma} = \sum_{c_j \in C} n^{\sigma}_{c_j}$ is the overall number of observations compatible with σ.
The associated credal set $\mathscr{P}^{\sigma}$ contains all probability distributions P on C, restricted by $p_j \in \left[\underline{P}^{\sigma}_{c_j}, \overline{P}^{\sigma}_{c_j}\right]$ for all $j = 1, \ldots, k$.
In Abellán and Moral (2003*b*) an easily computable algorithm to obtain the upper entropy distribution is presented, provided a set of probability intervals as obtained by the IDM. However the set is required to be proper and reachable.

Definition 8. *Proper and reachable set of probability intervals*[3]
A set of probability intervals $\{[l_i, u_i]\}_1^n$ is called proper, iff condition

$$\sum_{i=1}^{n} l_i \leq 1 \leq \sum_{i=1}^{n} u_i \tag{5.4}$$

[3]cf. De Campos et al. (1994), p. 168f

is fulfilled.
A proper set of probability intervals $\{[l_i, u_i]\}_1^n$ *is called reachable, iff conditions*

$$\sum_{j \neq i} l_j + u_i \leq 1 \tag{5.5}$$

and

$$\sum_{j \neq i} u_j + l_i \geq 1 \tag{5.6}$$

are met.

By the construction of the IDM the obtained posterior lower and upper probabilities define a proper and reachable set of probability intervals.[4]
As the constraints on the set of probability intervals are guaranteed to be fulfilled, Abellán and Moral's algorithm may be applied safely. Note that in case the observations assigned to the node contain missing values, the properness and reachability is natively obtained only under certain circumstances.[5]
Starting with the lower bounds of the intervals in the set under consideration, it recursively increases the value of those with minimal value, until the constraint of a probability distribution is reached.[6] [7]
Afterwards the Upper Entropy Distribution is employed to estimate the Upper Entropy as a plug–in estimator to $G(\mathscr{P})$ (See equation (5.2)), as generalized Shannon Entropy. The value of it characterizes the impurity in the node.
The purpose of splitting in a node is to decrease the impurity. As candidates for splitting are all those predictor variables which are not in the configuration defining the node. This approach is reasonable as the k–array splitting is applied: A predictor variable X previously used for splitting already restricts the observations in the node to $X = x_j$, with j being a fixed index characterizing any category in X, hence a further refinement regarding X is not achievable.
As the aim is an impurity reduction, for each splitting candidate the Upper Entropy is calculated as if it was already selected:
Let X be one splitting variable candidate with its states in $\{x_1, \ldots, x_J\}$. If the node with configuration σ is split according to X, J daughter nodes are created. For each of these daughter node its Upper Entropy $G(\mathscr{P}^{\sigma \cup (X=x_j)})$ is calculated. It follows the same steps as for the mother node, but with fewer observations taken into account due to the restriction $X = x_j$ being enforced. The Upper Entropies for each daughter nodes are then combined by summing them up, weighted proportional to its observations

$$G\left(\mathscr{P}^{\sigma \cup X}\right) = \sum_{j=1}^{J} \frac{n^{\sigma \cup (X=x_j)}}{n^{\sigma}} G(\mathscr{P}^{\sigma \cup (X=x_j)})\,. \tag{5.7}$$

The variable X^* with minimal $G\left(\mathscr{P}^{\sigma \cup X}\right)$ is then chosen as splitting variable, provided $G\left(\mathscr{P}^{\sigma \cup X^*}\right) < G\left(\mathscr{P}^{\sigma}\right)$. In that case for each state in X^* a daughter node is created and the splitting process is repeated on each of those. If no

[4] The proof is in Appendix B.1
[5] De Campos et al. (1994) provide a way to circumvent this, cf. page 169f (Proposition 2)
[6] As it is a discrete probability distribution the constraint is that the sum over all the states' probabilities equals 1.
[7] An algorithmic outline is given in Appendix A.2

variable is found to reduce the impurity or no splitting candidates are left, the node is declared as *leaf*.[8] Another termination criterion is the number of minimal observations assigned to any possible new daughter node. The less observations are in a daughter node, the broader are its posterior probability intervals generated by the IDM, inducing more imprecision thus making it more difficult to obtain dominated states of the classification variable, as will be seen in the next section 5.2. The experimental results on different data sets in Abellán and Moral (2003*a*) indicate that growing the trees to a maximum size ignoring a pre–specified minimal leaf size does not lead to overfitting as in context of precise classification trees. Yet in this master thesis such a minimal number of observations within a leaf is considered.
The initial splitting algorithm was extended in the sequent articles of Abellán and Moral,[9] in the way that not even a node's children are considered but also their grandchildren. However the latter will not be considered in here.
As Strobl (2005) pointed out the Upper Entropy is sensitive to different number of categories in the predictor variables, favouring those with a higher number of categories. However, the effect decreases with increasing number of observations. She proposes as reasonable alternatives Abellán and Moral's original measure of total uncertainty $TU(\mathscr{P}) = G(\mathscr{P}) + IG(\mathscr{P})$[10] and a correction to the estimation of the entropy, similar to the one introduced by Miller (1955)

$$\hat{H}_{\text{Miller}}(\hat{p}) = \hat{H}(\hat{p}) + \frac{|C| - 1}{2N}, \tag{5.8}$$

with N being the number of observations which are included in entropy–calculation and $|C|$ being the number of categories of the classification variable.
Regarding IG it is obtained by calculating the Möbius Inverse function of the power set $\mathscr{P}(C)$ first. As the lower probabilities are obtained by the IDM the Möbius Inverse function of any subsets of $\mathscr{P}(C)$ besides the singletons and the complete set reduces to 0 due to the additivity induced by the Dirichlet distribution. The values of the singletons coincide with the lower probability bounds and the value of the complete set is obtained by the restriction concerning the sum of all focal elements. However the final value of IG only consists on the Möbius Inverse function of the whole set multiplied with the logarithm of its cardinality, due to the singletons having cardinality 1 and all other subsets having a Möbius Inverse function's value of 0. The value of IG thus depends only on the number of categories (by $\ln(|C|)$) and the general interval width of the probability interval $\frac{s}{N+s}$, mainly specified by the number of observations. A simulation study performed by Strobl (2005) implies that the correction tends to be overcautious and only "*reliable for sufficiently large N and small $|C|$*".[11][12]

5.2 Decision in the leaves

In the previous section 5.1 a description of a general generation method of a tree structure was given. However the main interest in a classification tree is

[8] An algorithmic outline can be found in Appendix A.1
[9] e.g. Abellán and Moral (2003*a*), Abellán and Moral (2005)
[10] cf. Abellán and Moral (1999)
[11] Notation adapted; note from the author
[12] Strobl (2005), p. 7

not necessarily the underlying structure but the predictive ability. Each leaf is labelled with a single class it predicts in the case of precise classification trees. All observations attached to this leaf are then classified to the same value.
The general concept also applies to imprecise classification trees, but here a leaf is not restricted to predict one single state but rather any possible set of state. To account for the information residing in the observations assigned to the leaf any non-dominated state is returned as class prediction of the leaf.
One dominance criterion applied to credal set is *interval dominance*.[13] Zaffalon introduces also the *credal dominance*,[14] but the returned non–dominant states coincide as the credal set was generated by the IDM.

Definition 9. *Interval dominance*
Let C be a discrete random variable defined over $\mathscr{C}$ and let c' ; $c'' \subseteq \mathscr{C}$ be two generic events. Let X represent what is known, and let the probabilities $P(c'|X)$ and $P(c''|X)$ be, respectively, represented by the intervals $I' = [\underline{P}(c'|X); \overline{P}(c'|X)]$ and $I'' = [\underline{P}(c''|X); \overline{P}(c''|X)]$:
The interval I' is said to dominate I'' if

$$\underline{P}(c'|X) > \overline{P}(c''|X)\,;$$

in this case c' is said to interval dominate c''.

The rationale in the comparison is that the probability intervals create a partial order. After obtaining all non-dominated states of the classification variable, they are assigned to the corresponding leaf.[15]
As pointed out before, the number of observations in the leaf dramatically influences the number of predicted states in it. Recalling the estimation of the posterior probability intervals, the denominator consists of the number of observations under consideration N and the pre–chosen hyper-parameter s of the IDM. As seen in chapter 4, equation (4.3), for a fixed value s the interval width increases with smaller values N thus making it less likely to obtain dominated states. The same result is obtained when increasing the parameter s for fixed N. Especially at leaves with only a few observations assigned to it, proportional to the number of possible state $N/|C| \approx 3$, larger values of $s > 2$ tend to generate no dominated state.
When a new observation is to be classified, it is passed down the tree, starting from the root node to a leaf according to its values of the splitting variables on its way down. After reaching a leaf, its assigned non–dominated states are employed as prediction of the classification variable for the new observation.
Whenever an observation is predicted only one single state, this observation is said to be *determinate*, in all other cases, may it be 2 or more states, it is said to be *indeterminate*. Those two states will be important when measuring the accuracy of a tree.
Another approach, as in Abellán and Moral (2005), does not utilize the information the credal set provides: the *maximum frequency criterion*. The most frequent state of the classification variable is assigned to the leaf and employed for its prediction. It generates more determinate observations (leaves) in comparison to interval dominance which may be seen as advantageous. However in

[13] cf. Zaffalon (2002), p. 9
[14] cf. Zaffalon (2002), p. 14
[15] An intuitive algorithm is described in Appendix A.3

uncertain situations, i.e. one state beats all others by just one observation with a lot under consideration, a certainty is pretended, which is more accurately reflected by an indeterminate prediction derived from interval/strong dominance.

5.3 Measuring the performance of a credal classifier

To evaluate the predictive accuracy of a precise classification tree the misclassification rate is a reasonable measure. It gives the relative frequency of incorrect classified cases for a set of observations which were not used in the learning process. This may be accomplished by the means of pre–dividing the observations into learning and test data, or for cross–validation or bootstrapping the out–of–bag observations.
In a naive approach one may try to extend it to credal classifiers. When the tree classifier is completely determinate on all those observations, then the misclassification rate may be a reasonable choice, in all other cases it underestimates the actual error.
Let CT_V be a classification tree, which predicts all possible categories of the classification variable C. Such a classifier is called vacuous. The misclassification rate is obviously 0, yet this classifier in not desirable, as no new information is revealed.
Another deficiency of the misclassification rate is dealing with different cardinality predicted states, especially when comparing credal classifiers, as comparing the optimal precise classifier, i.e. the classifier which outputs all states correctly, to a credal classifier which outputs also all states correctly but for every observation a set of two states. Accounting for the misclassification rate both classifiers are equally desirable, yet it is obvious that the precise one should be preferred.
This observation leads to the concept of *discounted–accuracy*: Both the misclassification rate and the discounted accuracy are a 0-1-loss function in the observations. The major difference is the way they are aggregated over all observations: While the misclassification rate gives all equal weight ($1/n$ with n being the number of observations) the discounted–accuracy weights them according to the number of predicted states ($\frac{1}{n} \cdot \frac{1}{\#\text{ predicted states}}$).

Definition 10. *Discounted–Accuracy*[16]
Let $\mathscr{C} = \mathscr{C}_1, \ldots, \mathscr{C}_n$ be a credal classifier for n observations. Let $\mathrm{I}_{\mathscr{C}_i}(C_i)$ be the indicator function of prediction $\mathscr{C}_i$ and C_i the true class. The discounted–accuracy is then given by:

$$d\text{-}acc(\mathscr{C}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\mathrm{I}_{\mathscr{C}_i}(C_i)}{|\mathscr{C}_i|} \ .$$

Yet there is a certain arbitrariness in the choice of the *discount* for a correct classification. In the above definition the discount increases linearly in the cardinality of the predicted states. One may obtain different results when transforming it elsewise. Hence Corani and Zaffalon (2009) propose to use a Friedman rank

[16] cf. Corani and Benavoli (2010), p. 333

test to evaluate which of the credal classifiers performs better. In their article they employ it to compare two different classifiers. However in case of comparing multiple credal classifiers, as in an ensemble, this method leads to excessive dual-comparisons. Another notable advantage of the discounted–accuracy is that it aggregates the available information into a single number, with values in $[0, 1]$, thus allowing it to be interpreted as a percentage.
As Zaffalon et al. (2011) point out the discounted–accuracy does not distinguish between a vacuous and a random classifier, as they have the same expected predictive accuracy. They propose to include into the measure the variance of the prediction. This is accomplished by the means of specifying a concave utility function. In their binary simulation studies it gets evident that the utility based approach is performing better in comparison to the discounted–accuracy. However they emphasize on the fact in order to "*generate sensible results when using utility-based metrics, it is* fundamental *to carefully elicit the decision makers utility*".[17]
Less convincing, Abellán and Moral in their work deal with the case of indeterminate predictions. In order to compare their trees to precise tree based classification methods, they did not classify those instances, inducing "*a loss of some valuable information in certain situations (if for example we have a set with two non–dominated classes when the number of possible classes is 5)*".[18]
Other measures, as the *single-set accuracy* or the *set-accuracy*, may be used to quantify the predictive accuracy of a tree. Both employ the correct classification rate but with only determinate (indeterminate) observations under consideration. Another aspect is the average output size of the classifier. The *determinacy* gives the relative frequency of determinately predicted instances, whereas the *indeterminate output size* the average number of classes when predicting indeterminately. As Corani and Zaffalon (2008) stress, the indeterminate output size and the set-accuracy are only meaningful for non-binary classification variables.[19]

[17] Zaffalon et al. (2011), p. 410
[18] Abellán and Moral (2005), p. 250
[19] cp. Corani and Zaffalon (2008), p. 594

Chapter 6

Ensemble Trees

In the previous chapter 5 the construction of imprecise classification trees was outlined, now those trees are employed to construct ensembles. Before actually dealing with imprecise ensembles some basic concepts are described.

6.1 Introduction

Ensemble methods are a technique which constructs multiple instances of a *base learner* and aggregates them afterwards for prediction purposes. The rationale of ensemble methods is to induce a variance of the base learner on a given training setting, resulting in a reduction of the variance in the prediction on a test setting using the aggregate. At the first glance it sounds illogical, why a increase in variance of the base learner in a training set should decrease it in a test setting. However a greater variance in the learning sample means that the aggregate of the base learners is less sensitive to small changes in it. Ensemble methods share in a way the same idea as the IDM: in both the predictive performance of a precise basis, Dirichlet prior and base learner, is increased by adding uncertainty, set of Dirichlet priors, multiple instances of the base learner, respectively.
Generally there are no restrictions to the base learner, however for an ensemble of an already robust base learner the accuracy is not notably increased. A base learner may be any supervised learning technique, e.g. classification trees. The way aggregation is performed depends on the type of data but in terms of classification analyses a majority vote is carried out. The observation to classify is predicted for each tree, resulting in a *vote* for a class in each tree. In a majority vote the class with the most votes is then returned as the ensemble's prediction for the observation to classify.
Unfortunately the increase in accuracy is obtained on the sake of interpretability of the underlying structure. For a single classification tree the underlying classification structure is obvious and interpretable, however for an ensemble it is sacrificed, as each tree may still be interpreted, yet the final aggregation does not allow such.
There exists a variety of procedures on how to generate an ensemble. The most popular are bagging, random forests (Breiman (1996), Breiman (2001)) as well as the Boosting algorithms of Freund and Schapire (1966). Those will be revised

shortly and also the more robust approach of TWIX (Potapov (2009)) and an ensemble method employing imprecise trees on the basis of entropy ranges in the splitting process Crossman et al. (2011). All those previously listed will be looked into in terms of a classification task.

6.2 Bagging, Random Forests and Boosting

The method of Bagging (**b**ootstrap **agg**regating) was introduced by Breiman (1996). It generates classification trees on the basis of bootstrap samples of the learning data. Bühlmann and Yu (2002) gave a definition on the algorithmic nature of a bagging in terms of a regression task.[1] However this may be adapted without much effort to a classification set–up.

Definition 11. *Bagging classification trees*
Provided a set of data pairs $L = \{L_i\}_1^n, \quad L_i = (C_i, X_i)$ *with* C_i *being the classification variable and* X_i *a set of* p *feature attributes, the parameter of interest is the predicted class* $\hat{\theta}_C$*, which is a function* t *based on the learning sample* L*. So for any instance* x *the predicted class is obtained by,*

$$\hat{\theta}_C(x) = t_L(x) = t_{L_1,\ldots,L_n}(x).$$

Bagging is then defined in the following way:

1. *Generate a bootstrap sample* L^* *of* L*,*
2. *Compute the bootstrap predictor* $\hat{\theta}^*_C(x)$ *by the plug–in principle to* $\hat{\theta}^*_C(x) = t_{L^*}(x) = t_{L^*_1,\ldots,L^*_n}(x)$*,*
3. *The bagged predictor is then obtained to* $\hat{\theta}^*_{C;B}(x) = vote^* \left(\hat{\theta}^*_C\right)(x)$.[2]

As pointed out by Bühlmann and Yu (2002), the third step is calculated by repeatedly performing the first two steps, J times, and later voting over the J different results of $\hat{\theta}^*_{C;j}(x)$, $j \in \{1, \ldots, J\}$.
In their article they conclude that bagging smooths out instability, introduced by a hard decision, such as thresholds,[3] and thus lowering variance in the prediction.
However Breiman demonstrated that bagging classifiers may not necessarily lead to an improved classifier under majority voting.[4]
As Bagging includes bootstrapping the learning set, a test set may be constructed for each sample from the leftover, so called out–of–bag observations. This is advantageous in situations where the number of available observations is low. Breiman (1996) argues that even the complete learning set may be employed as test set.[5]
The random forests of Breiman (2001) are related to bagging. In each node the considered splitting variables are drawn at random. The randomization allows to grow an even broader variety of trees, thus increasing the predictive

[1] cf. Bühlmann and Yu (2002), p. 927f
[2] Note that *vote** may be any adequate calculus to obtain a dominating class
[3] i.e. Splitting performed according to values of an attribute variable
[4] cf. Breiman (1996), p. 130f
[5] cf. Breiman (1996), p. 131f

ability. While in Breiman (2001) only the feature variables are randomized, Dietterich (2000) advocates to draw a split point from a certain number of optimal attribute variables in the node. Quite obviously Bagging is included in the concept of random forests in case all feature variables are considered in the splitting process.
An advantage of random forests in comparison to bagging is that the correlation between feature variables in broken. So the researcher is able to identify attribute variables which are linked closely together and with almost equal predictive ability, whereas in the set–up of bagging as only the more dominant one would have been identified, thus providing a measure of feature variable importance along with.
While random forests (bagging as special case) grow trees on (sub)samples of the original data obtained by equally weighting them, boosting incorporates sampling according to weights. The most popular algorithm is AdaBoost by Freund and Schapire (1966). It re–weights the sampling probabilities for the next step according to the performance in the previous. Thus misclassified observations are assigned a greater weight whereas for correct classified ones their weight is decreased.
The performance of bagging, random forests and boosting of trees in comparison to other ensemble methods or single trees (other tree derivates) has been studied i.a. Breiman (1996), Bühlmann and Yu (2002), Dietterich (2000) and Gatnar (2008), focussing on different fusion rules. It appears that boosting is a reasonable method in a situation with low noise in the classification variable, while bagging and random forests outperform it for noisy set–ups.

6.3 TWIX and Ensemble trees under imprecise entropy

In the previous section 6.2 the ensemble is constructed by repeatedly growing a classification tree on a varying basis of observations. In the following, two methods are described where the ensemble is build in the actual tree growing process.
TWIX[6] (**T**rees **wi**th e**x**tra Splits) grows multiple trees by choosing not a single splitting point, but a pre–specified number m of the most favourable ones. The most favourable cutpoints could be either the m local maxima of the splitting criterion or m highest values of the criterion, or according to a grid. To avoid the problem of one variable shadowing others they allow to set m overall or per feature variable. Shadowing may be induced by varying numbers of categories in the feature variables[7] or a high correlation. The splitting procedure leads to a nested tree. To obtain a prediction either an optimal tree out of those nested, calculated by the means of cross–validation, or an aggregate of all trees may be applied.
Potapov et al. (2006) indicate that TWIX trees (ensembles) yield more accurate results on certain data. However this comes along with extremely high computational cost.[8]

[6] Potapov (2009)
[7] cp. chapter 5.1; Strobl (2005)
[8] cp. Potapov et al. (2006), p. 12

To tackle this difficulty, Strobl and Augustin (2009) presented a splitting procedure which identifies the robust cutpoints. They analyse it in case of a binary split tree, nonetheless conclude that it may be adapted to the non-binary case. As in each split two daughter nodes are generated, they identify robust cutpoints by assigning virtual observations to each of the daughter nodes and then recalculating the Gini gain, which they incorporate as the splitting criterion. The maximum number of virtual observations is chosen in advance. To obtain the Gini gain they employ the IDM and the upper entropy, although not in terms of a model decision but as a tool. The more virtual observations are added the more likely the splitting point will be different from the one obtained without any. They conclude that the minimal number of virtual observations required to change the initial cutpoint is a reasonable measure of the cutpoint's robustness. Similarly to TWIX trees prediction is accomplished by either predicting an optimal tree or the whole ensemble as aggregate.
Another approach of nested trees is presented in Crossman et al. (2011). In this, rather than using a single probability distribution to estimate the entropy, as in Abellán and Moral (2003*a*)[9], a set of entropies is compared to split in a node. This is accomplished by computing the distributions with maximal and minimal entropy, provided a credal set of probabilities. The credal set may be calculated by either the IDM or by a NPI[10] approach, as in Coolen et al. (2010) for ordinal data. An entropy interval is then calculated according to the *potential* and the *guarantee* of the credal set[11] being the minimum and maximum attained entropy for all distributions in it. The tree is then calculated as follows: For a given node compute the entropy interval for each of the splitting candidates. Select amongst the entropy intervals only those which are not dominated, their linked feature variables are then used for splitting. The complete tree including the node is then cloned as many times as there are splitting variables and one is assigned to each once. If one aggregates over all those grown trees with equal weight, one would favour those feature variables which introduce a higher number of sub–trees (i.e. more non–dominated entropy intervals). To tackle this the trees are weighted down in each cloning step to ensure that each mother node has the same weight.[12]
The two last approaches already introduced the employment of imprecise probabilities to obtain improved classification trees. In the next section 6.4 a reasonable approach of bagging imprecise classification trees is presented.

6.4 Bagging imprecise classification trees

The subject of bagging imprecise classification trees as described in chapter 5 has already been studied by Abellán and Masegosa (2010). In their article they create an ensemble of imprecise trees and compare it to bagging classification trees based on Quinlan's C4.5 trees. The ensemble was aggregated by majority vote. For a fair comparison they adapted their imprecise trees to deal with missing values and continuous feature variables.[13] They conclude that for data

[9] cp. chapter 5.1
[10] Nonparametric Predictive Inference
[11] cp. Crossman et al. (2011), p. 131
[12] compare the intuitively example in Crossman et al. (2011), p. 132
[13] cf. Abellán and Masegosa (2010), p. 253f

sets with medium–high classification noise bagging imprecise trees reduces the classification error.
However they do not give an explanation on the choice of majority vote. In the following an approach is presented why the majority vote is a reasonable choice of a fusion rule for imprecise classification trees.
The majority voting rule for an instance x over an ensemble of M classifiers is defined as

$$C(x) = \underset{c_j}{\operatorname{argmax}} \left(\sum_{m=1}^{M} \mathrm{I}\left(C_m(x) = c_j\right) \right),$$

where c_j are all classes of the classification variable C.
Without loss of generality assume the case of a binary classification variable. For a given observation the imprecise classification tree may predict three different outcomes C_1, C_2 or $\{C_1, C_2\}$. As the training data are bootstrapped for each classification tree their results are independent in the sense that they do not influence each other. In case a tree returns the indeterminate prediction $\{C_1, C_2\}$ it is considered as vote for both classes C_1 and C_2 when calculating the majority rule.
The attained predictive accuracy of majority rule is dependent on the "luck" of the researcher, however a slight improvement may be obtained in an average situation. A beneficial aspect of the majority rule is its ability to break ties between to classes. This becomes obvious when analysing certain types of observation within the classification task.
Easy to classify observations will still be identified by the majority vote as a majority of the trees will most likely identify them as such. Concerning those whose class label may be mistaken for another, i.e. "surrounded" by a different class, there is only a small chance that they are predicted correctly in each tree and thus in the majority vote. If there exists such a "lucky" tree, i.e. one that is correctly labelling those, a frequent occurrence in the ensemble even close to the majority is extremely unlikely. An improvement may be accomplished for those areas which are feasible to discriminate. These observations are most likely to be indeterminately classified, however some trees may label them determinately. In the aggregate those determinate trees decide for the predicted label.
On the one hand this behaviour is appreciated for those observations where there is a reason to believe that they should be classified determinately, on the other hand this leads to an extremely unsatisfactory result when all classification trees except one are unable to decide on a specific class and in the aggregation step this single tree decides on the final class.
To avoid such a situation the prediction of $\{C_1, C_2\}$ could be treated as a third class and not as vote for both classes. More generally each class in the power set of C is treated individually. In the case of a binary classification variable this approach seems reasonable for such an extreme situation as described above. Nonetheless it is a very conservative prediction rule as it results in an indeterminate output, when there are some trees, close to the majority, deciding determinately on the same class and the others are indeterminate. For more categories of the classification variable the number of possible ensemble output classes is increased drastically, leading yet to unsatisfactory results again: Consider the situation of m trees in the ensemble, which m being odd, where $\lfloor m/2 \rfloor$ trees vote for $\{C_1, C_2\}$ and the other $\lceil m/2 \rceil$ for $\{C_2, C_3\}$. Applying the original majority voting rule the ensemble predicts C_2, quite contrary the ma-

jority voting rule decides on $\{C_2, C_3\}$, ignoring the presence of C_2 in each tree's prediction.
In calculating the dominant class by majority voting all trees have been given equal weights. An alternative would be to assign weight to each tree according to its performance, thus weighting down those trees performing poorly on the provided data. However such an approach requires a sufficient number of observations in the data. A reasonable accuracy measure is one weighting the trees according to their discounted-accuracy.[14]
The weighted majority voting rule for an instance x over an ensemble of M classifiers is defined as

$$C(x) = \underset{c_j}{\operatorname{argmax}} \left(\sum_{m=1}^{M} \left(\mathrm{I}\left(C_m(x) = c_j\right) \cdot dacc(m)\right)\right),$$

where c_j are all classes of the classification variable C.
This measure takes only the performance of each tree on a given data set into account, so arbitrariness as previously described does still occur.
Up to this point only the predicted classes were considered. But the leaves of a tree also provide the attained posterior probabilities of the local IDM. Aggregating those over all trees in the ensemble may yield a sensible prediction of the ensemble.
The most conservative approach would be to get the union of all posterior probability distributions belonging to a certain observation. De Campos et al. (1994) provide the methodology to calculate it. The disjunction of a set of probability intervals is associated to the union of the according probability measures. "*The disjunction is the [conclusion; note from the author] inferred if at least one piece of observation is considered to be true.*"[15] They give the calculus of the disjunction $(l_1 \oplus l_2, u_1 \oplus u_2)$ for a pair of probability intervals $(l_1, u_1), (l_2, u_2)$ on a domain C as

$$(l_1 \oplus l_2)(A) = \min\left(l_1(A), l_2(A)\right), (u_1 \oplus u_2)(A) = \max\left(u_1(A), u_2(A)\right) \; \forall A \subseteq C \; . \tag{6.1}$$

This can be easily generalized to the disjunction of any number of probability intervals on the same domain.

Proposition 1. *Disjunction*
Let $\{I_j\}_1^n = \{(l_j, u_j)_1^n\}$ be a set of probability intervals on the same domain C. The disjunction $\bigoplus_j I_j$ is obtained to

$$\left(\bigoplus_j l_j, \bigoplus_j u_j \right)(A) = \left(\min_j(l_j(A)), \max_j(u_j(A)) \right) \; .$$

Proof. Due to the associativity of the disjunction operator, $\bigoplus_j I_j$ may be written to

$$\bigoplus_j I_j = (\ldots(((I_1 \oplus I_2) \oplus I_3) \oplus \ldots) \oplus I_n \; .$$

[14] compare chapter 5.3 for a discussion on accuracy measures
[15] De Campos et al. (1994), p. 176

This is then applied to all lower bounds:

$$\bigoplus_j l_j(A) = ((\ldots(((l_1 \oplus l_2) \oplus l_3) \oplus \ldots) \oplus l_n)(A) \ .$$

For the most inner expression the result is given in (6.1),

$$(l_1 \oplus l_2)(A) = \min\left(l_1(A), l_2(A)\right) \ .$$

The second inner most expression may be written as,

$$((l_1 \oplus l_2) \oplus l_3)(A) = \min\left(\min\left(l_1(A), l_2(A)\right), l_3(A)\right) = \min\left(l_1(A), l_2(A), l_3(A)\right) \ .$$

by replacing the expression in the inner brackets with the one above. Similarly are all outer brackets solved, resulting in

$$\bigoplus_j l_j(A) = \min_j(l_j(A))$$

Accordingly, the proof for the upper bounds is carried out. □

In case of a set of proper probability interval sets, the disjunction results in a proper set of probability measures. This guarantees that at least one probability distribution is defined by the intervals. Furthermore if reachable probability intervals are provided, the result is reachable as well.[16] In the general context interval probabilities, as in De Campos et al. (1994), they prove that the disjunction is not closed. However, this is only due to the way the disjunction operator handles non–singletons. As this master thesis does not discuss situations with such probability measures, the above probability measures are still called probability intervals.
To obtain the class(es) for prediction, the attained intervals are compared by any dominance criterion.[17] As the minimum (maximum) of the lower (upper) bound of the probability intervals is taken over all leaves, to which the observation to be classified belongs, the resulting intervals are decently wide. Thus interval dominated states are less likely to be seen at all. Especially in a noisy classification task this aggregation approach will most likely predict vacuously, i. e. all states.
Another method which is widely employed when combining the class probabilities of precise tree is the average (mean) rule.[18] For an ensemble of precise tree the result when applying it is indeed a single probability distribution, due to the commutativity of the addition. But does the average rule, when applied to an ensemble of imprecise trees, yield a proper or even reachable set of probability intervals?
As proven in Appendix B.3 the outputted set of probability intervals is reachable. Similarly to the disjunction rule and the decision in the leaves the final class is obtained by applying any dominance criterion on it. In comparison to the disjunction rule the width for the probability intervals is smaller as the average over all widths is taken, whereas the disjunction rule provides a maximal range. Moreover the average rule generates class probability intervals of the

[16] See Appendix B.2.1 and B.2.2
[17] See chapter 5.2
[18] cp. Gatnar (2008), p. 23

Table 6.1: Output of artificial binary experiment

	$\underline{P}(C=0)$	$\overline{P}(C=0)$	$\underline{P}(C=1)$	$\overline{P}(C=1)$
P_1	0.1	0.2	0.8	0.9
P_2	0.6	0.7	0.3	0.4
$av(T_1, T_2)$	0.35	0.45	0.55	0.65
$dis(T_1, T_2)$	0.1	0.7	0.3	0.9

same width in case the IDM is applied in the tree generation step. The last is even true for any general method resulting in class probability intervals of same width.[19] By calculating the mean, all trees get the same weight. As proposed for the majority vote, another weighting, according to some accuracy measures, may be beneficial, however this is not being studied herein.

When applying the disjunction rule to an ensemble, the output contains all original probability distributions, although broadening the range in comparison to a single tree. This does not hold for the average rule, as it centres the distributions. Assuming a binary case with the results according to Table 6.1, then the resulting sets of probability distributions are shown in Figure 6.1.

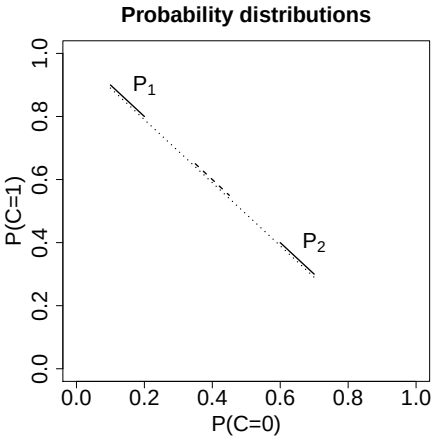


Figure 6.1: Probability distributions according to the binary result in Table 6.1: The dotted line $(\cdots)$ depicts all probability distributions in the disjunction of P_1 and P_2, while the dashed line $(---)$ is the resulting set of probability distributions by the average rule.

The probability distributions P_1 and P_2 are obviously disjunct, i.e. their intersection is empty.

This simple example contrasts the differences in the two decision rules. When applying the disjunction all trees in the ensemble are considered fully trustworthy, even most extreme ones, so no information on the outstanding situations is discarded. As pointed out previously this aggregation method is considerably sensitive to outliers. Just one tree in the ensemble is able to spoil an otherwise homogeneous result. The average does not fully trust all trees. It tries to manage the trees' opinion and thus discarding some probability distributions the trees offer, especially those of outliers. It adapts to a central probability distribution.

In the next chapter 7 the results of a simulation comparing the previously described aggregation methods and also the accuracy of the whole ensemble in contrast to a single tree are presented.

[19] See Appendix B.3.3

Chapter 7

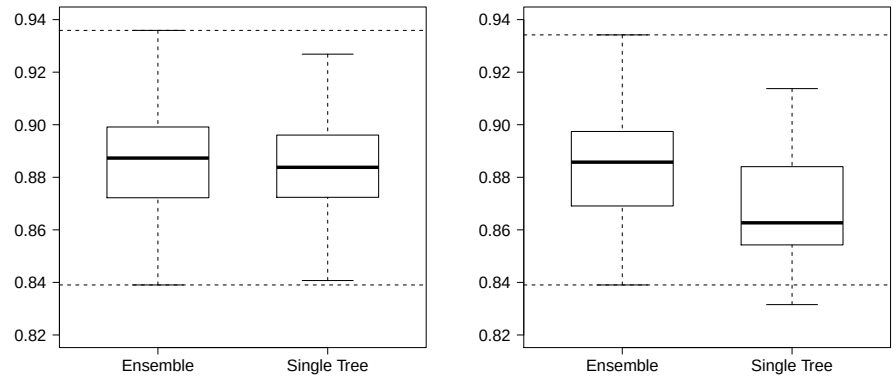
Simulation study

In order to study how the meta parameter s of the IDM influences a single tree and an ensemble, a simulation was performed on real data. The impact of the different aggregation methods as outlined in chapter 6.4 are compared on artificially generated data. They may be seen as a first indication on how the predictive ability is affected. The simulations carried out are not exhaustive and the presented results are subject to the underlying data. To obtain more generalizable results more different data should be considered, as in the simulations done only the concept is depicted.
All simulations are run using the statistical software *R*.[1] The tree building algorithm was implemented employing those algorithms described in Appendix A. The prediction of either a single tree or an ensemble is performed by a function written in *C*.[2]
At first the results on the real data set *SPECT Heart Data Set*[3] are described comparing a single imprecise tree with a bagged ensemble of imprecise trees concerning the behaviour subject to changes in the IDM parameter s and the stopping rule of minimal leaf size. In section 7.2 the different aggregation methods are studied on an artificial data set.

7.1 Simulation on SPECT Heart Data Set

The data set describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images, into a binary classification variable, 'normal' and 'abnormal'. The here employed data set has 22 binary feature attributes. Overall are 267 instances available which were divided into 80 instances on a training set and 187 in a test set, however this artificial split is ignored.
In order to reliably extract a potential difference in the behaviour of a bagged ensemble and a single tree, 50 bootstrap sample were generated on the complete data set (267 instances), which were employed as learning sets. The left–over observations of each were employed as test set to assess the predictive accuracy. Within each bootstrap sample both a single tree and an ensemble of 50 were constructed for different values of s and mls, as IDM parameter and minimal

[1] R Development Core Team (2012*a*)
[2] The code of it is based on the prediction function employed by TWIX (Potapov (2009))
[3] Frank and Asuncion (2010)



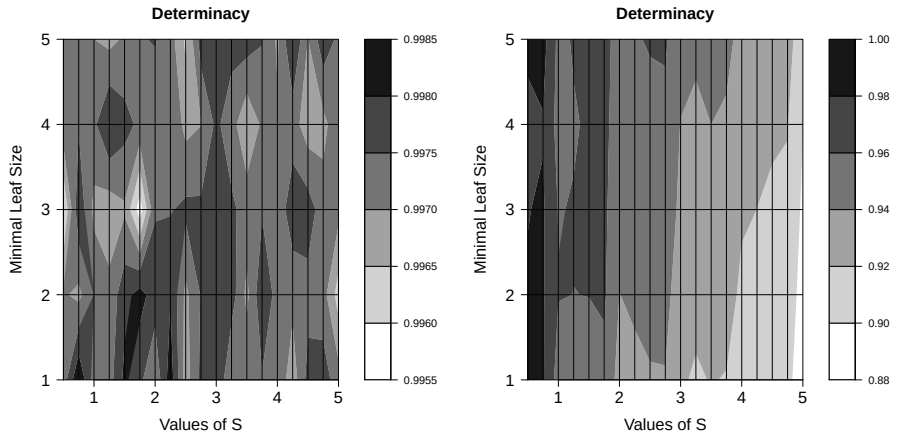
(a) Single–Set Accuracy (b) Discounted–Accuracy

Figure 7.1: Boxplots of Accuracy for optimal Bag and Single Tree

leaf size respectively. Regarding the minimal leaf size integer values from 1 to 5 (5 different values) were considered. For s the values were in the range between 0.5 and 5 (19 different values) in steps of 0.25. To identify a possible interaction they were varied concomitantly, thus resulting in a 5×19 matrix of 95 settings. To account for any structure in the bootstrapped sample, a 10–fold cross–validation was carried out in each setting. The setting's accuracy was calculated to the average accuracy obtained by the cross–validation, measured by the *determinacy*, *single–set accuracy* and *discounted–accuracy*. The set based measure are not considered, as the classification variable is binary. The prediction of the ensemble is obtained by the majority voting rule.
A direct comparison of the two different methods is applicable when considering the *best* classifier of the single trees and ensembles within each bootstrap sample. Due to the simulation's limits, induced by the settings, the *overall best* classifier, with respect to all hyper–parameters, may not be present in the sample, however the one with highest accuracy within it is a reasonable approximation.
The best model is assessed individually for the above mentioned accuracy measures. As depicted in Figure 7.1a there is only a slight difference regarding the single–set accuracy between the optimal ensemble tree and its single tree counterpart. This indicates that the single tree predicts as accurately as the ensemble on determinate leaves. However when looking at the discounted–accuracy (Figure 7.1b) the bagged classification trees attain significant[4] higher ones.
One reason might be that the bagged imprecise trees attain a significant greater determinacy,[5] thus less vacuous predictions, which have a less numerical effect on the discounted accuracy. When considering the single-set accuracy alongside, it means that those hard to classify instances are more determinately predicted in case of an ensemble. For almost half the bootstrap samples the discounted– and the single–set accuracy coincide for the bagged trees, whereas the discounted–accuracy is always smaller than the single–set accuracy for the single trees.

[4]Mann-Whitney-U test performed on a 5% significance level
[5]Mann-Whitney-U test performed on a 5% significance level

Condensely, the bagging of imprecise trees leads to an increase in the discounted–accuracy and determinacy in comparison to a single imprecise tree, but the single–set accuracy remains nearly unaffected.
The second aspect of this simulation is to compare the influence of the parameter s and the minimal leaf size mls. To obtain estimates for each setting, the average over all bootstrap samples is taken.
For a single imprecise tree one would expect the determinacy to decrease with lower values of s and mls, as the impurity within the leaves increases, induced by a more conservative class estimation (s) and a smaller size of the tree (mls).

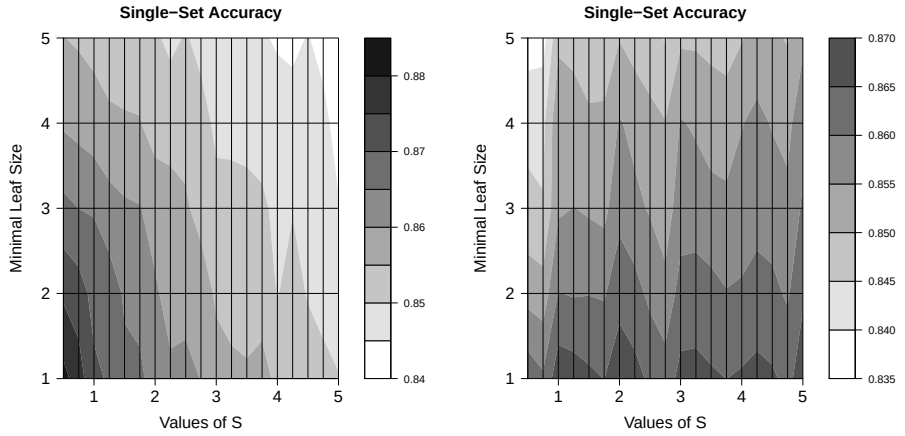


(a) Bagged Imprecise Trees (b) Single Imprecise Tree

Figure 7.2: Contour plot of average determinacy

Figure 7.2b[6] supports the intuitive approach in case of a single imprecise tree. It demonstrates that s has a greater impact on the determinacy as mls. However in the case of the bagging (Figure 7.2a) neither a clear and intuitive structure, nor a trend is visible. But on the second sight it appears that all coloured categories in Figure 7.2a are included in the upper most (dark grey) in Figure 7.2b. This supports the aforementioned superiority of the Bagged Imprecise trees in terms of determinacy.
As a tree grown with higher values of s has wider class probability intervals, it is likely to generate more leaves with indeterminate states (as seen in Figure 7.2b) but the accuracy of the remaining determinate leaves should not be affected. On contrast, the number of minimal observation mls is expected to influence the single set accuracy as it is an external stopping rule independent of the upper entropy in the node. In Figure 7.3b both intuitive assumptions are confirmed, yet concerning s a slight increase in single-set accuracy is visible for larger values of s. This may be accounted for larger values of s filtering out more difficult to predict instances, leaving only those in a single set which high evidence in the data.
Interestingly, for the bagged imprecise trees the assumption on the impact of s made on a single tree does not hold. As Figure 7.3a illustrates the single

[6]Caution should be taken when interpreting the contour plots: They are generated only by values on the intersection of grid lines, including the margins.

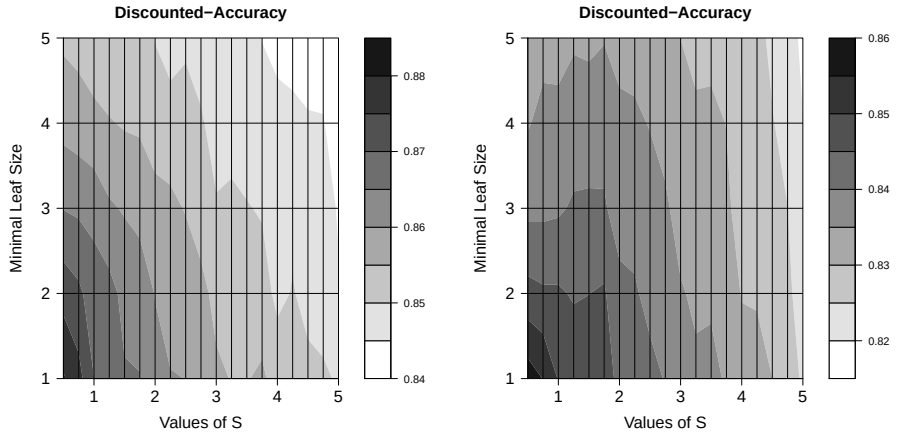


(a) Bagged Imprecise Trees (b) Single Imprecise Tree

Figure 7.3: Contour plot of average single–set accuracy

accuracy decreases with growing s. It may be explained by the aggregation rule applied. As the majority voting rule tends to produce a single class even when there is evidence that a set would be more appropriate, it is able to lead to false conclusions on the true class.[7] Whereas the single tree is invariant to changes in s the bag is receptive generating less accurate results for higher values of s. However the minimal leaf size has the same effect as on a single tree.
Concerning the range of attained single–set accuracy, neither the bag nor the single tree should be favoured as their accuracies are in almost the same range, when considering all configurations.
As the classification is binary the discounted–accuracy is mainly affected by the single–set accuracy and the determinacy, thus the previous results are combinable to form hypotheses about the discounted–accuracy. For a single tree it is reasonable to assume that the highest value will be for small s and mls as both the determinacy and the single–set accuracy are high in that area, whereas for larger values of both the discounted–accuracy should decrease. Figure 7.4b shows indeed such behaviour.
Similarly the result for an ensemble is expected, yet from a different background. As seen in Figure 7.2a the determinacy is almost 1 for each configuration, which means for the discounted–accuracy of the bag that it will be mostly influenced by the single–set accuracy. Indeed the effect of s and mls in Figure 7.4a, depicting the average discounted–accuracy of the ensemble, is almost equal to the single–set accuracy (Figure 7.3a).
In direct comparison of the attained discounted–accuracy of the ensemble and the single tree they are almost of the same shape but the values are generally higher in case of the ensemble.
Overall the simulation on this particular data set indicates that the discounted–accuracy, as a naive measure of the general predictive ability, is affected by both the artificial stopping rule mls and the parameter s of the IDM. For both the discounted–accuracy decreases with higher values. As mls was introduced in

[7] cp. chapter 6.4



(a) Bagged Imprecise Trees (b) Single Imprecise Tree

Figure 7.4: Contour plot of average discounted–accuracy

order to avoid overfitting as in a precise classification context, the above results indicate that in case of imprecise classification trees, either single or bagged, it may be neglected as the best result on the test sets is obtained for its overall minimum value 1.
Furthermore, the simulation has demonstrated that the favour for slower values of s is justified. In case of the SPECT data set the smallest value of $s = 0.5$ attained the optimal results regarding the bag, both in case of the single–set and the discounted–accuracy. However in case of the single tree it does not hold for the single–set accuracy as a slight favour for larger values was found.
Nonetheless the simulation covers only a small aspect of all classification tasks. The results pointed out here should be viewed as an indication on the effect of s and the arbitrary stopping rule, but not as strong evidence or even proof. To obtain more general statements more classification settings on a variety of different data sets must be carried out in a much broader simulation.
In the following, the results of a simulation comparing the different aggregation methods are described.

7.2 Simulation on artificial data

In the simulation the differences in the aggregation/fusion methods are of main interest. The ensemble is formed by bagging imprecise classification trees. The class prediction dependent rules *majority voting* and *weighted majority voting*, as well as methods based on the aggregation of probability intervals, *disjunction* and *average rule* were studied. As the latter ones only provide probability intervals as result, the actual classes were predicted by the *interval dominance* and the *maximum frequency criterion*. Thus 6 different aggregation methods are competing. In order to maintain a base line a single imprecise tree is grown alongside. The predictive ability is evaluated by *determinacy*, *single–set accuracy* and *discounted–accuracy*.
The data set employed contains a binary classification variable and 10 feature

attributes on 1050 instances. The class variable C is drawn from a Bernoulli distribution with equal chances for the two classes. In order to obtain class dependent feature variables X, a restriction on the conditional chances was compelled: $|\pi_0 - \pi_1| \geq 0.1$, with $\pi_j = \mathbb{P}(X = 1|C = j)$.[8] This restriction ensures that both conditional chances are not too similar or even equal. The π_j are drawn form a uniform distribution on $[0, 1]$. If the restriction was not satisfied, both were discarded. π was sampled 50 times and for each π 100 data sets were generated, thus overall 5000 different data sets. The π are classified according to the difference in their conditional chances: data sets created by π with a small absolute difference (≤ 0.4) are considered as having a high classification noise, as the class dependencies are less strong; absolute difference values between 0.4 and 0.7 are of medium, and greater as 0.7 of low classification noise. This allows for a more detailed comparison of the fusion methods. In the simulation were 4 π associated with low noise, 15 with medium and 31 with high noise.
On each data set a bag of 50 imprecise trees and a single one is grown on the first 50 instances, the remaining 1000 were employed as test set to assess the accuracy. According to the results in the previous section 7.1, the minimal leaf size and s were globally set to 1.
At first, the determinacy of the different aggregation techniques is presented as it allows to deduce some aspects of the discounted–accuracy. When comparing the dominance criterion within an aggregation rule, one would expect the interval dominance generate less determinate outputs than the maximum frequency criterion.

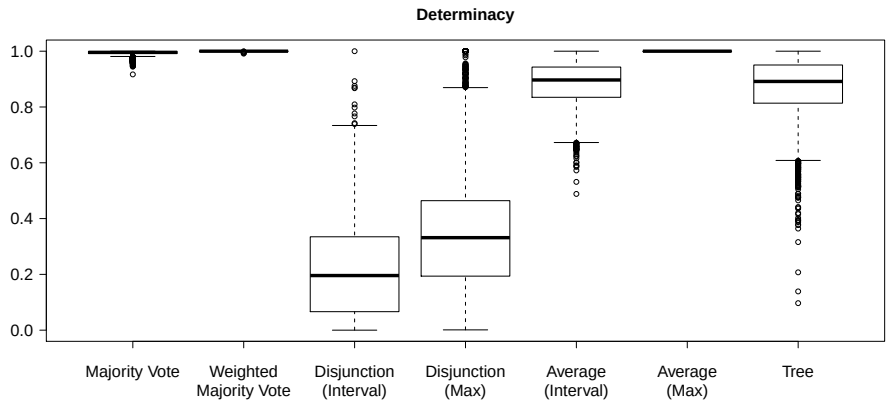


Figure 7.5: Boxplot of determinacy over 5000 test sets

Figure 7.5 gives the attained determinacy over the 5000 test sets for the different methods. Both majority voting rules lead to almost exclusively determinate prediction results. Such a behaviour as already suspected as described in chapter 6.4. The graphic also supports the aforementioned hypothesis concerning the different dominance criteria. The average rule in disjunction with strong dominance attains not as high determinacy as the majority voting rules, but

[8]The data generation is analogously to Zaffalon et al. (2011)

is on the same level as the single tree. However when applying the maximum frequency criterion the average rule generates just determinate outputs, reaching the highest possible determinacy of all methods under consideration. The least determinate outputs are generated by the disjunction rule. Even when applying the maximum frequency criterion, still 75% of the ensembles attained a determinacy of lower than 50%. This is due to a lot of vacuous aggregated predictions obtained by the disjunction rule, before applying a dominance criterion. The conclusions remain the same when distinguishing between the classification noise induced by π.[9]
As the single–set accuracy is the other factor influencing the discounted–accuracy, those results are presented next.

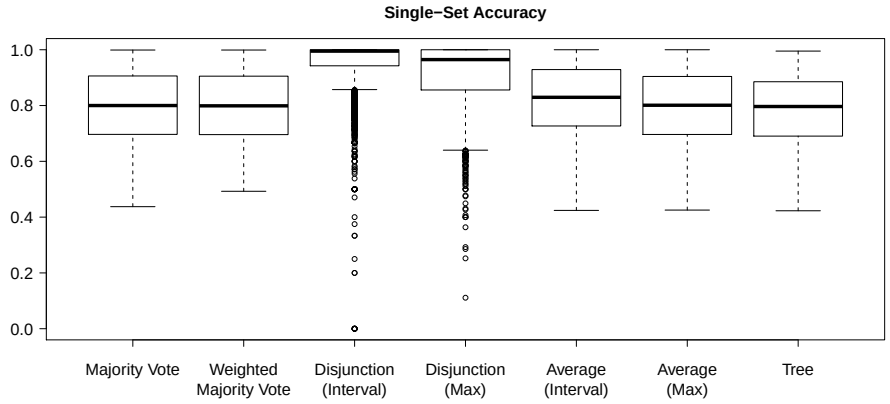


Figure 7.6: Boxplot of single–set accuracy over 5000 test sets (4689 for *disjunction (Strong)*)

At first it should be noted, that for 311 test sets of high classification noise the classifier applying the disjunction rule together with interval dominance was completely vacuous, so a single–set accuracy could not be assessed. Besides the disjunction rule, the methods differ only slightly with the weighted majority rule showing the smallest range as it is visible in Figure 7.6. Where for the determinacy the tree was inferior to the aggregation methods (excluding the disjunction rule), concerning the single–set accuracy it is on the same level. Regarding the single–set accuracy of the disjunction rule, it is superior to all other methods. As the disjunction rule was introduced as a somewhat conservative approach, it seems justified, because it classifies only those instances determinately when there is strong evidence for a certain class. When comparing the different classifying criteria, in terms of single–set accuracy the interval dominance has a significant[10] greater mean than the maximum frequency for both the average and the disjunction rule. When considering the classification noise, the disjunction rule generates by far the most precise outputs, as well as applying the interval dominance criterion.[11]

[9] See Appendix C.1
[10] Mann-Whitney-U test performed on a 5% significance level
[11] See Appendix C.2

The simulation demonstrates that neither determinacy nor single–set accuracy may be taken as a measure for overall accuracy, as they lead to quite opposing results, yet capturing only one aspect of the classifier. In the following the discounted–accuracy is considered, merging the previous conclusions.

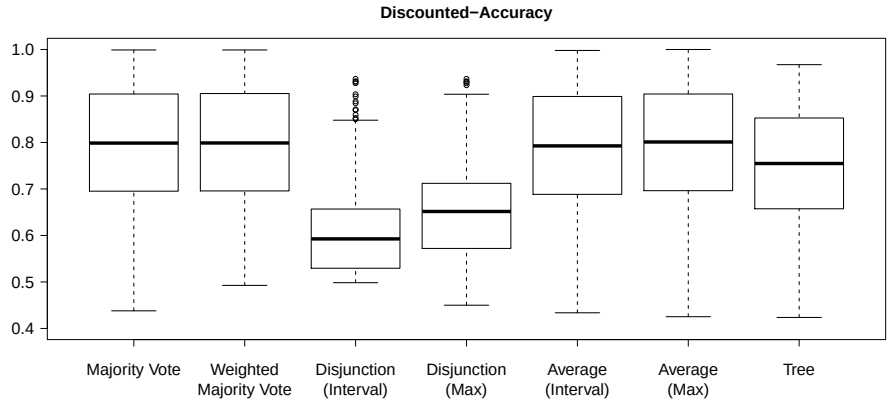


Figure 7.7: Boxplot of discounted–accuracy over 5000 test sets

Both majority voting rule based aggregation methods attained the same median in discounted–accuracy, differing only in terms of the minimal reached value, which is higher for the weighted vote (Figure 7.7). Due to the weighting being based on the discounted–accuracy of each tree in the ensemble, the increased lower value is not surprising, however for more than 75% of the test sets there is no difference in those 2 rules. As the determinacy of the disjunction rules is comparably low overall, the discounted–accuracy reflects it, leading to the overall lowest medians. When considering the average rules, there is little difference and they are on the same level as the majority voting based aggregation methods. In comparison to the discounted–accuracy of the single tree both the majority voting and the average rules have a significant greater median.[12] For both the disjunction and the average rule the interval dominance results in a significant lower median of discounted–accuracy.
The results remain the same when accounting for different classification noise in the data, yet for more noisy settings the difference between the average and majority voting rule and the disjunction rule is smaller.[13]
Considering all above results it seems that there is only little difference in the majority voting based fusion methods. In terms of discounted– and single–set accuracy they are equal, only for determinacy the weighted outperforms the standard majority voting. As the weighted majority voting assesses the accuracy of each tree in the ensemble, the additional computational effort required seems not to be satisfied.
As already stated at the introduction of the disjunction rule the extremely wide probability intervals lead to less determinate outputs. However when single–set accuracy is the main concern they provide the most accurate outputs, yet for

[12]Mann-Whitney-U test performed on a 5% significance level
[13]See Appendix C.3

only a few instances. In any other cases the disjunction rule is considerably conservative, sometimes even vacuous.
When talking about the average rule the dominance criterion for the final class prediction needs to be considered. While the average rule with maximum frequency criterion attains the highest determinacy it comes at cost of single–set accuracy. When applying the interval dominance instead, those effects are balanced, yet the discounted–accuracy attained is still significantly lower.
Comparing the dominance criterion applied to the disjunction and average rule, the maximum frequency criterion yielded more determinate outputs than the interval dominance, but less accurate on those. As main advantage of interval dominance appears its general superiority in terms of single–set, whereas the maximum frequency criterion attains a higher determinacy and discounted–accuracy.
The deployment of ensemble seems justified when looking at the achieved accuracy of the single trees: For both the discounted– and the single–set accuracy the ensemble aggregated by the majority and the average rule, independently of weighting or dominance criterion, attains higher values than the simple tree. Also when exclusively considering the single–set accuracy the disjunction rule is superior to the single tree as the single trees' achieved single–set accuracy is significant lower than the one of the ensemble.

Chapter 8

Conclusions and further research

In this master thesis the statistical background of imprecise classification trees is outlined. Based on the concept of Dempster–Shafer theory they allow to account for imprecision in the learning data set, yet attaining a reasonable degree of predictive accuracy. Contrary to precise classification trees they are more robust to changes in the learning data. The algorithm by Abellán and Moral (2003*a*) is based on Quinlan's ID3 algorithm employing an entropy splitting criterion. The class probabilities are estimated as relative frequencies by the so called Imprecise Dirichlet Model to probability intervals.
The Imprecise Dirichlet Model is based on the choice of the parameter s, which affects the introduced imprecision. In literature a value of $s = 1$ or $s = 2$ is commonly applied, but this is an arbitrary choice. In their work on imprecise trees, Abellán and Moral set the value for s to 1, but gave no empirical justification for such a choice. An analysis, performed on the *SPECT Heart Data Set*,[1] revealed that the actual choice of s has a great effect on the attained accuracy, favouring lower values of s for the determinacy and discounted–accuracy. However, regarding the single–set accuracy, it slightly increases with greater values of s.
The tree–growing method by Abellán and Moral (2003*a*) does not include any stopping criterion, such as a minimal leaf size. Herein the effect of a minimal leaf size was considered, affecting the discounted–accuracy through the single–set accuracy. In the analysis on the SPECT data set, the highest accuracy was achieved for a value of 1, equalling no restriction.
In a next step the bagging of imprecise trees was considered. Bagging those trees has already been studied by Abellán and Masegosa (2010), but with the main purpose of comparing the imprecise to precise bags. Similarly to imprecise trees the behaviour of the imprecise bag, when considering different values of s and a minimal leaf size was evaluated on the SPECT data set, leading to almost the same conclusions: The restriction on the minimal leaf size is unnecessary and lower values of s give higher accuracy. In direct comparison of the two classification techniques, bag of imprecise trees versus a single tree, bagging attained a significant higher median in discounted–accuracy.

[1] Frank and Asuncion (2010)

As the above conclusions are based on just one data set, they may be misleading. To further support or confute those, a broader variety of different data sets needs to be taken into account. The influence on the accuracy subject to the number of trees within an ensemble is also worth studying.
The creation of ensembles automatically raises the question on how to aggregate their outputs into a single one. In Abellán and Masegosa (2010) the majority voting rule is employed. The simulation on an artificial data set in chapter 7.2 supports the choice. A weighted majority rule, with weights based on the attained discounted–accuracy for each tree, does not notably improve the predictive accuracy. Regarding probability based fusion rules, the disjunction and the average rule were considered. As they output probability intervals, in a second step the outputted classes need to be estimated by a dominance criterion. The disjunction is extremely cautious and sometimes even vacuous when classifying, but on determinately predicted instances, it outperforms all considered aggregation methods. The average rule however is rather similar to majority voting, but in conjunction with the maximum frequency criterion the most determinate. For the sake of simplicity the classification variable was binary in both simulations. One step to generalize the above results could be to expose the methods to a non-binary classification. Then accuracy measures as the *set–accuracy* should be considered.
The studied aggregation rules employ only the properties of probability intervals, however in general they are not limited to them. In the context of a non-binary classification task the interval probability, induced by the probability intervals,[2] may be combined as in (Troffaes, 2006), by assigning *trust* to each tree. However for large ensembles the calculation gets feasible as the author stated himself.[3] From De Cooman and Troffaes (2004) another aggregation rule is deducible which collapses to the disjunction rule in the case of conflict in the trees,[4] but with a large ensemble this is likely to appear.
Further modifications may be made when changing the underlying model to estimate the class probabilities, as in Crossman et al. (2011) the ordinal NPI for an ordinal classification variable.
As already stated in the introduction, the feature variables were limited to categorical ones. As this is arbitrary, the tree growing algorithm should be adapted to deal even with continuous attribute variables in a similar fashion as Quinlan's C4.5 algorithm.[5]
Another generalization of the aggregation/fusion would be to allow missing values. In the current state the algorithm is capable of dealing with missing values in the training set on both the feature and the classification variable by ignoring them in the locally applied IDM in each node. However, missing values in any feature variable in the test set are not allowed, thus they do not influence any aggregation.

[2] In De Campos et al. (1994) a general method on how to obtain interval probabilities when applying the disjunction is described (p. 178)
[3] cp. Troffaes (2006), p. 378f
[4] The solution to problem 3 may be seen as the aggregation rule.
[5] This algorithm was employed as basic tree inducer in Abellán and Masegosa (2010)

Appendix A

Algorithms

A.1 Tree growing algorithm

In the following a algorithmic version of the tree growing process is given:[1]

Algorithm 1 Tree growing algorithm

Input:	A node containing observations and a configuration
Output:	A tree structure with knots and leaves

Initialization: $\mathscr{L} \leftarrow \{X_i\}_1^n$

$TreeBuild(No, \mathscr{L})\,\{$
  if $(\mathscr{L} = \emptyset)$ **then** {**exit**}
  $\sigma \leftarrow$ configuration of No
  Compute the Upper Entropy of No: $\alpha_0 = G\left(\mathscr{P}^{\sigma}\right)$
  Compute $\alpha = \min_{X \in \mathscr{L}} G\left(\mathscr{P}^{\sigma \cup X}\right)$
  if $(\alpha \geq \alpha_0)$ **then** {
    exit # Making No a leaf
  } **else** {
    Let X^* be the variable for which the minimum α is attained
    Remove X^* from $\mathscr{L}$
    Assign X^* to Node No
    for (State $x_k^* \in$ {States of X^*}) **do** {
      Add a Node No_k
      Make No_k child of No
      Call $TreeBuild\left(No_k, \mathscr{L}\right)$
    }
  }
}

[1] It is based on the algorithm *BuildTree* given in Abellán and Moral (2005) on page 246; adapted to the case where just children are taken into account.

A.2 Upper Entropy Algorithm

In the following the Abellán and Moral's algorithm to calculate the Upper Entropy distribution is described:[2] [3]

Algorithm 2 Upper Entropy Algorithm

Input:	A set of reachable probability intervals $[l_i, u_i]_n^1$
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
Imin(x, S):	returns the index/indices of the minimum value of the array x considering only indices in S
Sig(x, S):	return the index/indices of second minor value of the array x considering only indices in S; if not existent returns -1
Nmin(x, S):	returns the number of indices attaining the minimum value of array x considering only indices in S
Min(x, y, z):	return the minimum of values x, y, z

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

$GetMaxEntropy(l, u, \hat{p}, S)\{$
  for $(i = 1$ **to** $n)$ **do** $\{\hat{p}_i \leftarrow l_i\}$
  if $(Sum(l) < 1)$ **then** $\{$
    for $(i = 1$ **to** $n)$ **do** $\{$
      if $(l_i = u_i)$ **then** $\{$
        $S \leftarrow S - \{i\}$
      $\}$
    $\}$
    $s \leftarrow Sum(l)$
    $r \leftarrow Imin(l, S)$
    $f \leftarrow Sig(l, S)$
    $m \leftarrow Nmin(l, S)$
    for $(i = 1$ **to** $n)$ **do** $\{$
      if $(i \in r)$ **then** $\{$
        if $(f = -1)$ **then** $\{$
          $l_i \leftarrow l_i + Min(u_i - l_i, \frac{1-s}{m}, 1)$
        $\}$ **else** $\{$
          $l_i \leftarrow l_i + Min(u_i - l_i, l_f - l_r, \frac{1-s}{m})$
        $\}$
      $\}$
    $\}$
    $GetMaxEntropy(l, u, \hat{p}, S)$
  $\}$
$\}$

[2] cf. Abellán and Moral (2003*b*), p. 593f
[3] For a more intuitive understanding the notation is slightly modified.

A.3 Class Predicting Algorithms

Provided with a set of probability intervals these short algorithms removes the states according to interval dominance[4] and maximum frequency:

Algorithm 3 Class Predicting Algorithm applying Interval Dominance

Input: A set of probability intervals $\{[l_i; u_i]\}_1^n$
with a set of associated states $\mathscr{C} = \{c_1, \dots c_n\}$
Output: A set of non–dominated states $\mathscr{C}^*$

Initialization: $\mathscr{C}^* \leftarrow \{c_1, \dots, c_n\}$ # List of states

```
PredictClassIntDom ({[l_i; u_i]}_1^n) {
    for (i = 1 to n) do {
        for (j = 1 to n; j ≠ i) do {
            if (u_i < l_j) then {
                Remove c_i from C*
                Break j-loop and continue with i-loop
            }
        }
    }
}
```

Algorithm 4 Class Predicting Algorithm applying Maximum Frequency

Input: A set of probability intervals $\{[l_i; u_i]\}_1^n$
with a set of associated states $\mathscr{C} = \{c_1, \dots c_n\}$
Output: A set of non–dominated states $\mathscr{C}^*$

Initialization: $\mathscr{C}^* \leftarrow \emptyset$
$maxv \leftarrow -1$ # Stores the maximum value

```
PredictClassMaxFreq ({[l_i; u_i]}_1^n) {
    for (i = 1 to n) do {
        if (maxv = u_i) then {
            Add c_i to C*
        } else if (maxv < u_i) then {
            C* ← ∅
            Add c_i to C*
            maxv ← u_i
        }
    }
}
```

[4] cp. Corani and Zaffalon (2009), p. 31

Appendix B

Proofs

B.1 IDM generating proper and reachable sets of probability intervals

For sake of simplicity a classification variable C with 3 different states (C_1, C_2, C_3) is assumed. The number of observations under consideration is n_1, n_2 and n_3 for states C_1, C_2 and C_3 respectively. As the states are exhaustive the number of the observation in the different states sum up to the overall number n under consideration. Given an IDM with $s > 0$ one obtains the following intervals:

$$\text{For state } C_1: \quad I_1 = \left[\tfrac{n_1}{n+s}, \tfrac{n_1+s}{n+s}\right] \tag{B.1}$$

$$\text{For state } C_2: \quad I_2 = \left[\tfrac{n_2}{n+s}, \tfrac{n_2+s}{n+s}\right] \tag{B.2}$$

$$\text{For state } C_3: \quad I_3 = \left[\tfrac{n_3}{n+s}, \tfrac{n_3+s}{n+s}\right] \tag{B.3}$$

The set of probability intervals $I = \{I_1, I_2, I_3\}$ is proper:

Proof.
First summing up the lower bounds of the intervals

$$\begin{aligned}\sum_{i=1}^{3} l_i &= \frac{n_1}{n+s} + \frac{n_2}{n+s} + \frac{n_3}{n+s} \\ &= \frac{n_1+n_2+n_3}{n+s} \\ &= \frac{n}{n+s} < 1 \quad ,\end{aligned}$$

then the upper bounds

$$\begin{aligned}\sum_{i=1}^{3} u_i &= \frac{n_1+s}{n+s}+\frac{n_2+s}{n+s}+\frac{n_3+s}{n+s}\\ &= \frac{n_1+s+n_2+s+n_3+s}{n+s}\\ &= \frac{n+3s}{n+s}\\ &= 1+\frac{2s}{n+s}>1 \quad .\end{aligned}$$

□

The set I is reachable:

Proof.
Starting with the first Interval I_1:

$$\begin{aligned}l_2+l_3+u_1 &= \frac{n_2}{n+s}+\frac{n_3}{n+s}+\frac{n_1+s}{n+s}=\frac{n_1+s+n_2+n_3}{n+s}=1\leq 1\\ u_2+u_3+l_1 &= \frac{n_2+s}{n+s}+\frac{n_3+s}{n+s}+\frac{n_1}{n+s}=\frac{n_1+n_2+n_3+2s}{n+s}>1\end{aligned}$$

Due to the symmetric nature of the IDM regarding categories the result is obviously identical for I_2 and I_3. □

B.2 Properties of the disjunction for probability intervals

Without loss of generality a set of m sets of probability intervals $\left\{\{I_{ij}\}_1^k\right\}_1^m$ on a domain with dimension k, with $I_{ij}=(l_{ij},u_{ij})$. The disjunction leads to the result

$$\bigoplus_j I_{ij}=\left(\min_j(l_{ij}),\max_j(u_{ij})\right)=(l_i^*,u_i^*)\quad \forall i=1,\ldots,k\ .$$

B.2.1 The disjunction of proper probability intervals is proper

For now assume that each of the m sets is proper.

Proof.
At first the requirement to lower bounds is looked into:
By supposition all sets of probability intervals are proper:

$$\sum_{i=1}^{k} l_{ij}\leq 1\quad \forall j=1,\ldots,m\ .$$

Summing over all lower bounds of the disjunction leads to

$$\begin{aligned}
\sum_{i=1}^{k}\left(\min_{j} l_{ij}\right) &\leq \sum_{i=1}^{k}\left(\frac{1}{m}\sum_{j=1}^{m} l_{ij}\right) \\
&\leq \frac{1}{m}\sum_{i=1}^{k}\sum_{j=1}^{m} l_{ij} \\
&\leq \frac{1}{m}\sum_{j=1}^{m}\left(\sum_{i=1}^{k} l_{ij}\right) \\
&\leq \frac{1}{m}\sum_{j=1}^{m} 1 \\
&\leq 1
\end{aligned}$$

Accordingly is the proof of the restriction on the upper bounds. □

B.2.2 The disjunction of reachable probability intervals is reachable

By supposition the m sets are reachable this time.
As reachable includes proper, with the above proof it is evident that the result also yields a proper set of probability intervals. In the following it is proved that it is even reachable, i.e. equation (5.5) and (5.6) (chapter 5) are met.

Proof. By supposition (5.5) is valid for all sets, so especially for the one with $l_i^* = \min_j(l_{ij})$.

$$\begin{aligned}
1 &\leq \sum_{j\neq i} u_j^* + l_i^* = \sum_{j\neq i} u_j^* + \min_r(l_{ir}) \quad \forall i \\
&\leq \max_r\left(\sum_{j\neq i} u_{jr}\right) + \min_r(l_{ir}) \quad \forall i \\
&\leq \sum_{j\neq i}\left(\max_r(u_{jr})\right) + \min_r(l_{ir}) \quad \forall i
\end{aligned}$$

The last expression gives (5.5) for the disjunction result.
As (5.6) is valid for all sets, so especially for the one with $u_i^* = \max_j(u_{ij})$.

$$\begin{aligned}
1 &\geq \sum_{j\neq i} l_j^* + u_i^* = \sum_{j\neq i} l_j^* + \max_r(u_{ir}) \quad \forall i \\
&\geq \min_r\left(\sum_{j\neq i} l_{jr}\right) + \max_r(u_{ir}) \quad \forall i \\
&\geq \sum_{j\neq i}\left(\min_r(l_{jr})\right) + \max_r(u_{ir}) \quad \forall i,
\end{aligned}$$

demonstrating (5.6) for the disjunction result. □

B.3 Properties of the average rule for probability intervals

Without loss of generality a set of m sets of probability intervals $\left\{\{I_{ij}\}_1^k\right\}_1^m$ on a domain with dimension k, with $I_{ij} = (l_{ij}, u_{ij})$. The average rule leads to the result

$$Mean(I_{ij}) = \left(\frac{1}{m}\sum_{j=1}^{m}(l_{ij}), \frac{1}{m}\sum_{j=1}^{m}(u_{ij})\right) = (l_i^a, u_i^a) \quad \forall i = 1, \ldots, k \; .$$

B.3.1 The average over a set of proper probability intervals is proper

Proof.
Concerning the lower bounds, by supposition all sets are proper, i.e.

$$\sum_{i=1}^{k} l_{ij} \leq 1 \quad \forall j = 1, \ldots, m \; .$$

Does this also hold for the average rule's result?

$$\begin{aligned}
\sum_{i=1}^{k}\left(\frac{1}{m}\sum_{j=1}^{m}(l_{ij})\right) &= \frac{1}{m}\sum_{i=1}^{k}\sum_{j=1}^{m} l_{ij} \\
&= \frac{1}{m}\sum_{j=1}^{m}\left(\sum_{i=1}^{k} l_{ij}\right) \\
&\leq \frac{1}{m}\sum_{j=1}^{m} 1 \\
&\leq 1 \; .
\end{aligned}$$

Analogously is the proof on the restriction of the upper bounds. □

B.3.2 The average over a set of reachable probability intervals is reachable

Proof.
As reachability conditions on properness, from the above proof the result of the average rule is proper in any case.
For any $i \in \{1, \ldots, k\}$ the properties (5.5) and (5.6) are proven. Starting with (5.6):

$$\begin{aligned}
\sum_{j\neq i}\left(\frac{1}{m}\sum_{r=1}^{m} u_{rj}\right) + \frac{1}{m}\sum_{r=1}^{m} l_{ri} &= \frac{1}{m}\sum_{r=1}^{m}\left(\sum_{j\neq i} u_{rj}\right) + \frac{1}{m}\sum_{r=1}^{m} l_{ri} \\
&= \frac{1}{m}\sum_{r=1}^{m}\left(\sum_{j\neq i} u_{rj} + l_{ri}\right) \\
&\geq \frac{1}{m}\sum_{r=1}^{m} 1 = 1
\end{aligned}$$

Analogously for (5.5):

$$\begin{aligned}
\sum_{j\neq i}\left(\frac{1}{m}\sum_{r=1}^{m} l_{rj}\right)+\frac{1}{m}\sum_{r=1}^{m} u_{ri} &= \frac{1}{m}\sum_{r=1}^{m}\left(\sum_{j\neq i} l_{rj}\right)+\frac{1}{m}\sum_{r=1}^{m} u_{ri} \\
&= \frac{1}{m}\sum_{r=1}^{m}\left(\sum_{j\neq i} l_{rj}+u_{ri}\right) \\
&\leq \frac{1}{m}\sum_{r=1}^{m} 1 = 1
\end{aligned}$$

□

B.3.3 The average rule returns intervals of equal width for each class

Suppose every probability interval within each of the m sets has the same width. This allows to re-define the upper bound with respect to the lower bound and a width, fixed for each of the m sets:

$$u_{ij} = l_{ij} + d_j \quad d_j \geq 0 \quad \forall i \in \{1,\ldots,k\}, j \in \{1,\ldots,m\}$$

For any $i \in \{1,\ldots,k\}$ the width of the aggregated result is obtained to:

$$\begin{aligned}
w_i^a = u_i^a - l_i^a &= \frac{1}{m}\sum_{j=1}^{m}(u_{ij}) - \frac{1}{m}\sum_{j=1}^{m}(l_{ij}) \\
&= \frac{1}{m}\sum_{j=1}^{m}(l_{ij}+d_j) - \frac{1}{m}\sum_{j=1}^{m}(l_{ij}) \\
&= \cancel{\frac{1}{m}\sum_{j=1}^{m}(l_{ij})} + \frac{1}{m}\sum_{j=1}^{m}(d_j) - \cancel{\frac{1}{m}\sum_{j=1}^{m}(l_{ij})} \\
&= \frac{1}{m}\sum_{j=1}^{m}(d_j)
\end{aligned}$$

As seen from the last expression the width is independent of i and thus is is the same for all $i \in \{1,\ldots,k\}$, which proves the statement of equal widths.

Appendix C

Figures of the second simulation

The following boxplots depict the attained *determinacy*, *single–set accuracy* and *discounted–accuracy*, when distinguishing the tests sets by their classification noise induced in the data generation.

C.1 Determinacy

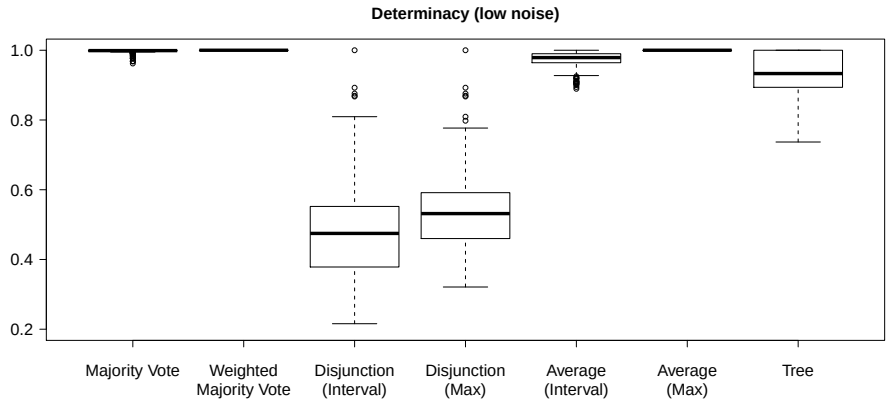


Figure C.1: Boxplot of determinacy over 400 test sets with low classification noise

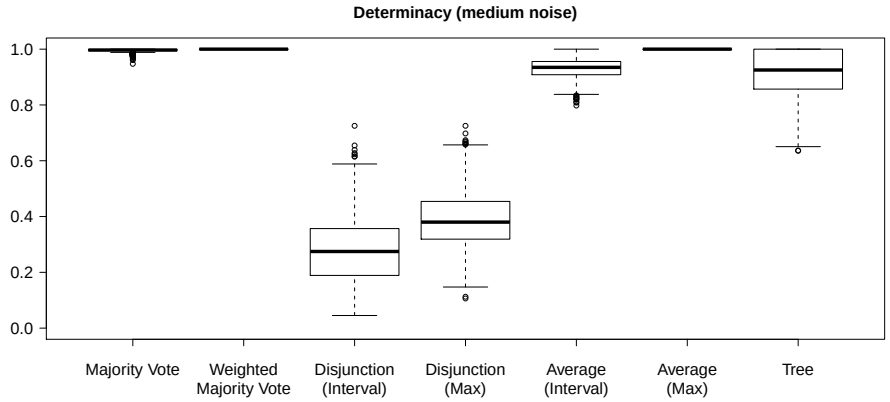


Figure C.2: Boxplot of determinacy over 1500 test sets with medium classification noise

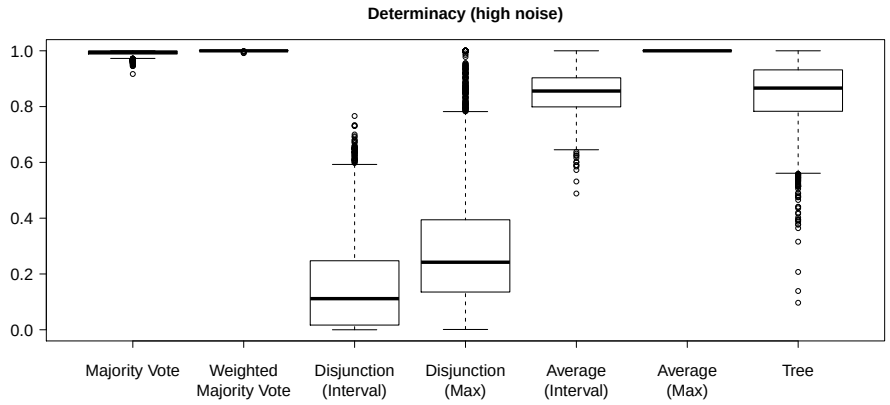


Figure C.3: Boxplot of determinacy over 3100 test sets with high classification noise

C.2 Single–set Accuracy

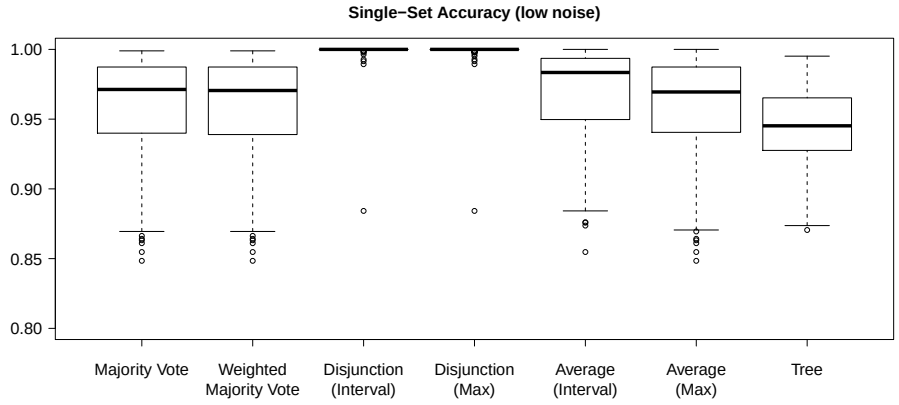


Figure C.4: Boxplot of single–set accuracy over 400 test sets with low classification noise

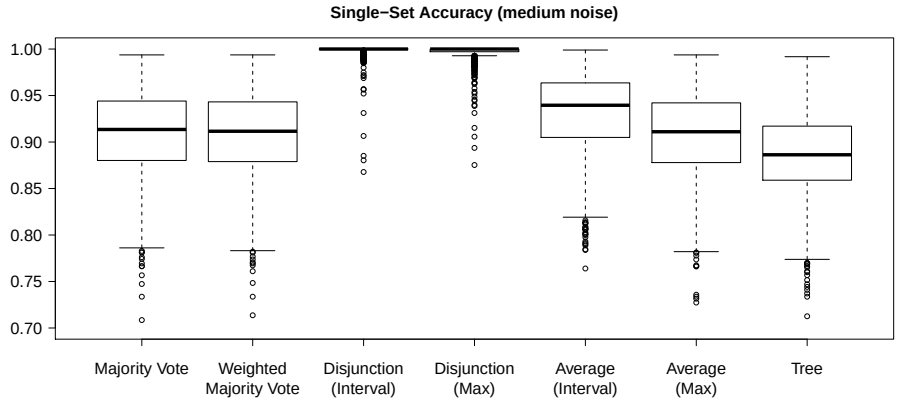


Figure C.5: Boxplot of single–set accuracy over 1500 test sets with medium classification noise

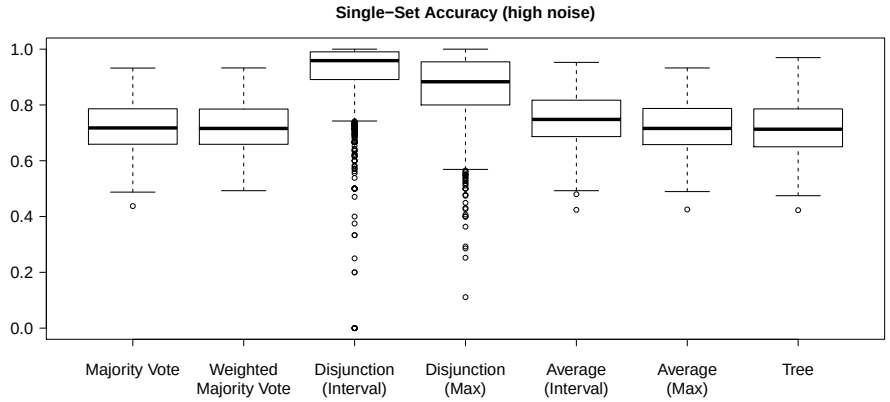


Figure C.6: Boxplot of single–set accuracy over 3100 test sets with high classification noise (2799 for *Conjunction (Strong)*)

C.3 Discounted–Accuracy

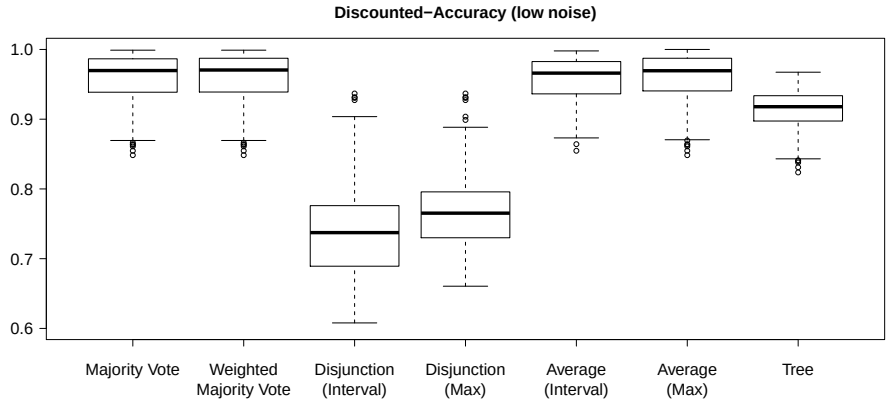


Figure C.7: Boxplot of discounted–accuracy over 400 test sets with low classification noise

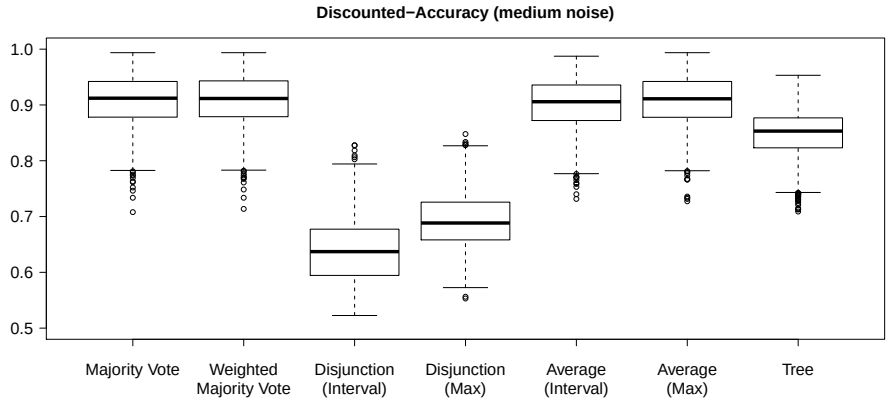


Figure C.8: Boxplot of discounted–accuracy over 1500 test sets with medium classification noise

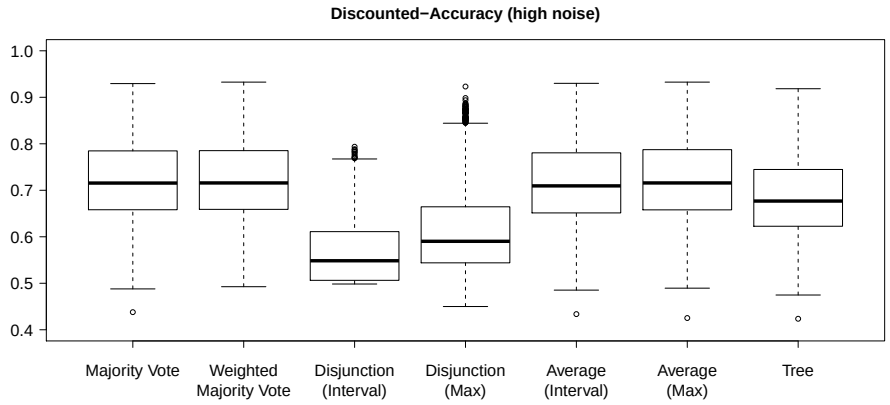


Figure C.9: Boxplot of discounted–accuracy over 3100 test sets with high classification noise

Appendix D

Attachment on electronic mediums

Along with this master thesis a CD is attached, containing the R[1]–scripts and C–sources, as well as the generated graphics and the final results of the simulations. Below the functions in the scripts are shortly summarized. A more detailed description could be found in the scripts itself. For full functionality the C–sources are required to be compiled by R.[2]

D.1 R–Scripts

accuracy.r: In this file the function *accuracy* is defined. It calculates the determinacy, single–set accuracy, average number of classes when indeterminate, set–accuracy and discounted–accuracy and return also the underlying prediction result.

treebuilder.r: This file contains the main functions of *imptree* and *impbags* which generate a single imprecise classification tree (object of class *imptree*) and a bag of imprecise classification trees (object of class *impbag*), respectively. Also the function *predclass* is defined in there, which predicts the attained classes based on a set of probability intervals.

predict.r: In this file the prediction functions for objects of class *imptree* and *impbag* are defined. For the bags one may specify different fusion rules such as *majority voting* and *weighted majority voting*, as well as *disjunction* and *average rule* with class prediction by *interval dominance* and *maximum frequency criterion*.

print.r: Herein functions could be found, defining a fancy printing for objects of class *imptree* and *impbag*.

init.r: This file sources all necessary files mentioned above in order to use all functions described above. It also loads the shared objects, generated by the C–sources.

[1] R Development Core Team (2012*a*)

[2] For more details on how to compile C–sources see R Development Core Team (2012*b*), chapter 5.5.

spectsimu.r: Program code of the simulation in chapter 7.1.

artificialsimu.r: Program code of the simulation in chapter 7.2.

D.2 C–Sources

maxentropy.c: Herein the function carrying out the calculation of the Upper Entropy Distribution is defined.

predict.c: In this file the prediction of observations for an imprecise classification tree is carried out, i.e the climbing down of the tree. The functions defined herein are based on those defined in *predict.c* of the TWIX-package.[3]

D.3 Outline of the contents

In the main directory the following items could found:

thesis.pdf: This master thesis in pdf-format.

R: The directory containing all R–scripts (Appendix D.1) and directories with results of the simulations, including a README-file each.

src: The directory containing all C–sources (Appendix D.2).

[3] Potapov (2009)

Bibliography

Abellán, J. and A. Masegosa (2010), Bagging Decision Trees on Data Sets with Classification Noise, *in* S.Link and H.Prade, eds, 'Foundations of Information and Knowledge Systems', Vol. 5956 of *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, pp. 248–265.

Abellán, J. and S. Moral (1999), 'Completing a Total Uncertainty Measure in the Dempster–Shafer Theory', *International Journal of General Systems* **28**(4-5), 299–314.

Abellán, J. and S. Moral (2000), 'A Non–Specificity Measure for Convex Sets of Probability Distributions', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **8**(3), 357–368.

Abellán, J. and S. Moral (2003*a*), 'Building classification trees using the total uncertainty criterion', *International Journal of Intelligent Systems* **18**(12), 1215–1225.

Abellán, J. and S. Moral (2003*b*), 'Maximum of entropy for credal sets', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **11**, 587–597.

Abellán, J. and S. Moral (2003*c*), Maximum of entropy in credal classification, *in* J. M.Bernard, T.Seidenfeld and M.Zaffalon, eds, 'ISIPTA'03: Proceedings of the Third International Symposium on Imprecise Probabilities and Their Applications', Vol. 18 of *Proceedings in Informatics*, Carleton Scientific, Lugano, Switzerland, pp. 1–15.

Abellán, J. and S. Moral (2005), 'Upper entropy of credal sets. Applications to credal classification', *International Journal of Approximate Reasoning* **39**, 235–255.

Bernard, J. M. (2005), 'An introduction to the imprecise Dirichlet model for multinomial data', *International Journal of Approximate Reasoning* **39**(2-3), 123–150. 3rd International Symposium on Imprecise Probabilities and Their Applications, Lugano, Switzerland, Jul 14, 2003.

Breiman, L. (1996), 'Bagging predictors', *Machine Learning* **24**(2), 123–140.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Breiman, Leo, J. H. Friedman, R. A. Olshen and C. J. Stone (1984), *Classification and Regression Trees*, Chapman and Hall/CRC, Boca Raton.

Bühlmann, P. and B. Yu (2002), 'Analyzing bagging', *The Annals of Statistics* **30**(4), 927–961.

Coolen, F. P. A., P. Coolen-Schrijner and T. A. Maturi (2010), On Nonparametric Predictive Inference for Ordinal Data, *in* E.Hüllermeier, R.Kruse and F.Hoffmann, eds, 'Computational Intelligence for Knowledge-Based Systems Design', Vol. 6178 of *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, pp. 188–197.

Corani, G. and A. Benavoli (2010), Restricting the IDM for Classification, *in* E.Hüllermeier, R.Kruse and F.Hoffmann, eds, 'IPMU (1)', Vol. 80 of *Communications in Computer and Information Science*, Springer, pp. 328–337.

Corani, G. and M. Zaffalon (2008), 'Learning Reliable Classifiers From Small or Incomplete Data Sets: The Naive Credal Classifier 2', *Journal of Machine Learning Research* **9**, 581–621.

Corani, G. and M. Zaffalon (2009), Lazy naive credal classifier, *in* J.Pei, L.Getoor and A.de Keijzer, eds, 'KDD Workshop on Knowledge Discovery from Uncertain Data', ACM, pp. 30–37.

Crossman, Richard J., J. Abellán, T. Augustin and F. P. A. Coolen (2011), Building Imprecise Classification Trees With Entropy Ranges, *in* F.Coolen, G.de Cooman, T.Fetz and M.Oberguggenberger, eds, 'ISIPTA'11: Proceedings of the Seventh International Symposium on Imprecise Probability: Theories and Applications', SIPTA, Innsbruck, pp. 129–138.

De Campos, L. M., J. F. Huete and S. Moral (1994), 'Probability Intervals: A tool for uncertain reasoning', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **2**(2), 167–196.

De Cooman, G. and M. C. M. Troffaes (2004), 'Coherent lower previsions in systems modelling: products and aggregation rules', *Reliability Engineering System Safety* **85**(1–3), 113–134.

Dempster, A. P. (1967), 'Upper and lower probabilities induced by a multivalued mapping.', *Annals of Mathematical Statistics* **38**(2), 325–339.

Dietterich, T. G. (2000), 'An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization', *Machine Learning* **40**, 139–157.

Dubois, D. and H. Prade (1985), 'A Note on Measures of Specificity for Fuzzy Sets', *International Journal of General Systems* **10**(4), 279–283.

Frank, A. and A. Asuncion (2010), 'UCI Machine Learning Repository'.
URL: *http://archive.ics.uci.edu/ml*

Freund, Y. and R. E. Schapire (1966), Experiments with a New Boosting Algorithm, *in* L.Saitta, ed., 'Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96)', Morgan Kaufmann, pp. 148–156.

Gatnar, E. (2008), Fusion of Multiple Statistical Classifiers, *in* C.Preisach, H.Burkhardt, L.Schmidt-Thieme and R.Decker, eds, 'Data Analysis, Machine Learning and Applications', Studies in Classification, Data Analysis, and Knowledge Organization, Springer, Berlin/Heidelberg, pp. 19–27.

Harmanec, D. and G. J. Klir (1994), 'Measuring Total Uncertainty in Dempster–Shafer Theory: A Novel Approach', *International Journal of General Systems* **22**(4), 405–419.

Maeda, Y. and H. Ichihashi (1993), 'An Uncertainty measure with monotonicity under the Random Set Inclusion', *International Journal of General Systems* **21**(4), 379–392.

Miller, G A (1955), 'Note on the bias of information estimates', *Information Theory in Psychology Problems and Methods IIB* **2**, 95–100.

Potapov, S. (2009), *TWIX: Trees WIth eXtra splits*. R package version 0.2.10.
URL: *http://CRAN.R-project.org/package=TWIX*

Potapov, S., M. Theus and S. Urbanek (2006), 'TWIX: Trees WIth EXtra Splits'. Presentation slides from the 3rd Ensemble Workshop in Munich 2006.

Quinlan, J. R. (1986), 'Induction of Decision Trees', *Machine Learning* **1**(1), 81–106.

Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers Inc.

R Development Core Team (2012*a*), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
URL: *http://www.R-project.org/*

R Development Core Team (2012*b*), *Writing R Extensions*, R Foundation for Statistical Computing, Vienna, Austria. ISBN: 3-900051-11-9.
URL: *http://www.R-project.org/*

Shafer, G. (1976), *A Mathematical Theory of Evidence*, Princeton University Press, Princeton.

Strobl, C. (2005), Variable Selection in Classification Trees Based on Imprecise Probabilities, *in* F. G.Cozman, R.Nau and T.Seidenfeld, eds, 'ISIPTA'05: Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications', SIPTA, Carnegie Mellon University, Pittsburgh, pp. 339–348.

Strobl, C. and T. Augustin (2009), 'Adaptive Selection of Extra CutpointsTowards Reconciling Robustness and Interpretability in Classification Trees', *Journal of Statistical Theory and Practice* **3**(1), 119–135.

Troffaes, M. C. M. (2006), 'Generalizing the conjunction rule for aggregating conflicting expert opinions', *International Journal of Intelligent Systems* **21**(3), 361–380. 6th Workshop on Uncertainty Processing, Hejnice, Czech Republic Sep 24-27, 2003.

Walley, P. (1991), *Statistical Reasoning with Imprecise Probabilities*, Statistics/Probability Series, Chapman and Hall, London.

Walley, P. (1996), 'Inferences from multinomial data: Learning about a bag of marbles', *Journal of the Royal Statistical Society Series B-Methodological* **58**(1), 3–34.

Yager, R. (1983), 'Entropy and Specificity in a Mathematical Theory of Evidence', *International Journal of General Systems* **9**(4), 249–260.

Zaffalon, M. (2002), 'The naive credal classifier', *Journal of Statistical Planning and Inference* **105**(1), 5–21.

Zaffalon, M., G. Corani and D. Mauá (2011), Utility-Based Accuracy Measures to Empirically Evaluate Credal Classifiers, *in* F.Coolen, G.de Cooman, T.Fetz and M.Oberguggenberger, eds, 'ISIPTA'11: Proceedings of the Seventh International Symposium on Imprecise Probability: Theories and Applications', SIPTA, Innsbruck, pp. 401–410.

Zaffalon, M., K. Wesnes and P. Petrini (2003), 'Reliable diagnoses of dementia by the naive credal classifier inferred from incomplete cognitive data', *Artificial Intelligence in Medicine* **29**(1–2), 61–79.

Affidavit

I, Paul Fink, hereby declare that this master thesis in question was written single–handed and no further as the denounced resources and sources were employed.

Munich, in May 2012

(Paul Fink)

Eidesstattliche Erklärung

Hiermit versichere ich, Paul Fink, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, Mai 2012

(Paul Fink)