Monotone Classification with Decision Trees

Christophe Marsala[1] **Davide Petturiti**[2]

[1]LIP6, Université Pierre et Marie Curie, Paris 6, 4 place Jussieu, 75005 Paris, France
[2]Dip. S.B.A.I., "La Sapienza" Università di Roma, via Scarpa 16, 00161 Roma, Italy

Abstract

In machine learning, monotone classification is concerned with a classification function to learn in order to guarantee a kind of monotonicity of the class with respect to attribute values. In this paper, we focus on rank discrimination measures to be used in decision tree induction, i.e., functions able to measure the discrimination power of an attribute with respect to the class taking into account the monotonicity of the class with respect to the attribute. Three new measures are studied in detail and an experimental analysis is also provided, comparing the proposed approach with other well-known monotone and non-monotone classifiers in terms of classification accuracy.

Keywords: Decision tree induction. Monotone classification. Rank discrimination measures.

1. Introduction

In machine learning, decision tree induction enables the construction of a summarized view of a set of data from a given training set.

Usually, basic approaches to construct a decision tree from a training set are based on a *Top Down Induction of Decision Tree* (TDIDT) method. A tree is built from its root to its leaves, by successive partitioning of the training set into subsets. An attribute is selected thanks to a *discrimination measure* H (in classical decision trees, the Shannon entropy is generally used [5, 26]) that ranks the attributes according to their discriminating power with regard to the class. The attribute with the highest discriminating power is selected to split the training set. Methods to construct decision trees differ mainly in their choice of H [20, 21].

In detail, at each step of the construction of a decision tree, the measure of discrimination H is used to value the discrimination power of each attribute with regard to the class. Thus, it will produce a ranking of all the attributes according to this value, and the *winner* attribute will be the one that is ranked first (i.e., the one that has the lowest value). As a consequence the whole ranking is not interesting in this process (only the first one is selected).

In the fuzzy setting, fuzzy decision trees (FDTs) have been extensively used in the past years as a powerful knowledge extraction tool, and nowadays they are still an active domain of researches and applications [1, 8, 9, 19, 24, 29]. Very recent works have also shown that FDTs can be used also in ranking applications where it is more useful to associate test examples with a degree of classification rather than a crisp class [17].

Indeed, there exist a lot of real-world application problems where the values of the class are symbolic and ordered. In that kind of problems, it appears that finding attributes that are *gradually* linked with the class could be more valuable in order to explain the decision process done by means of that tree. For instance, *the older the patient, the most vulnerable to disease.*

Commonly, fuzzy TDIDT algorithms are obtained via a fuzzification of a discrimination measure. However, classical (fuzzy) discrimination measures [5, 26, 32] take into account only the informative properties of attributes with regard to the class and forget to handle the graduality that could link their values.

Here, as a preparatory step to fuzzification, we focus on the crisp case. Thus we consider a training set of objects ω_i's described by attributes a_j's, each ranging in a totally ordered set X_j, and labelled with a class coming also from a totally ordered set C. More formally, the *monotone classification problem* (see, e.g., [25]) consists in finding an order preserving extension λ' defined on the description space X generated by the X_j's, of a monotone consistent labelling function λ specified on a set of object descriptions $E \subseteq X$.

Anyway, real data are generally neither monotone consistent nor consistent, i.e., λ could not be monotone on E or worse λ could be only a relation on $E \times C$.

In the literature, some monotone classifiers have been proposed [2, 3, 7, 6, 18, 11, 27] but, as shown in [4], they deeply suffer from non-monotone noise present in the data and in many cases do not have classification accuracy as primary goal. Moreover, in order to ensure the monotonicity of the final classifier λ', a monotonization phase of the initial dataset or of the final classifier could be necessary [10, 25], causing a possible loss of information.

In particular, the global monotonicity constraint acts on the final classifier λ' and so it requires an a priori knowledge of the entire tree. Hence, global monotonicity is difficult to enforce in an inductive construction procedure since at each step only one attribute can be taken into account.

Our aim is to inductively build a decision tree exploiting somehow the eventual monotonicity present in the dataset, anyway, since no assumption of monotonicity is made on the data, we need to relax requirements on the final dataset.

This is why, in our approach, we adopt a greedy strategy: at each step of the building process we choose the attribute a_j "enforcing the most" the *local monotonicity constraint*, i.e., for all $\omega_i, \omega_h \in \Omega$,

$$a_j(\omega_i) \leq a_j(\omega_h) \Rightarrow \lambda(\omega_i) \leq \lambda(\omega_h).$$

As a consequence it is not possible to expect a globally monotone classifier in the end. Nevertheless, we seek at least a weak form of monotonicity in the case the given dataset is monotone consistent.

In order to build a monotone decision tree, new kinds of discrimination measures should be used. More precisely, measures able to quantify the monotonicity of λ w.r.t. a_j and being robust to non-monotone noise are required in such a process.

In [15] the authors propose a rank generalization of Shannon mutual information, namely *rank mutual information*, which is a combination of Shannon entropy with *dominance rough set* relation [12, 13], based on the object-wise writing of Shannon entropy. In the same paper they underline that this measure is both sensitive to monotonicity and robust to noisy data. In [14] this measure is used to build binary tree classifiers guaranteed to possess a weak form of monotonicity (*rule monotonicity*) in the case the starting dataset is monotone consistent. They call this TDIDT algorithm REMT and show it behaves well compared to both monotone and non-monotone classifiers. Moreover, in [16] they use the rank mutual information for feature selection.

In [23], we applied the same rank generalization procedure given in [15] to other two deeply studied discrimination measures such as Gini measure and Yuan and Shaw measure, moreover we introduced directly a third measure not having a non-monotone counterpart. A detailed study of aforementioned measures has been carried on in [22].

In order to show the effectiveness of the proposed measures we wrote a binary tree classifier parametrized by a rank discrimination measure H^*. This TDIDT algorithm, called RDMT(H^*), is based on REMT [14] and is written in Java using the WEKA package. RDMT(H^*) has been tested on artificial and real datasets and compared with other well-known monotone and non-monotone classifiers in terms of classification accuracy. Our analysis shows our classifier can exploit the eventual monotonicity of the dataset: it can compete with non-monotone classifiers in accuracy and, moreover, it is much more robust to non-monotone noise than purely monotone classifiers.

The paper is organized as follows. In Section 2, we present some new discrimination measures for attribute ranking that enable to take into account a monotone link between a descriptive attribute and the class. In Section 3, a new binary decision tree classifier is proposed that exploits the proposed rank discrimination measures. In Section 4, a set of experiments is presented in order to compare our proposed algorithm to existing ones and to highlight better its properties. Finally, we conclude focusing on fuzzification of the proposed measures as a natural expansion of present work.

2. Rank discrimination measures

Let us consider a set $\Omega = \{\omega_1, \dots, \omega_n\}$ of *objects* or *alternatives* described by a family $\mathcal{A} = \{a_1, \dots, a_m\}$ of *attributes* with finite totally ordered range (also called *true criteria* in [6, 7]), i.e., for each $j = 1, \dots, m$, a_j is a function on Ω ranging in $X_j = \{x_{j_1}, \dots, x_{j_{t_j}}\}$ with $t_j > 1$ and $(X_j, \leq)$ totally ordered. We assume a *labelling function* $\lambda : \Omega \to C$ is given, where $C = \{c_1, \dots, c_k\}$ is a set of *classes* with $k > 1$ and $(C, \leq)$ also totally ordered.

We stress that, for $i = 1, \dots, n$, each object ω_i can be mapped to a corresponding $(m+1)$-tuple $(a_1(\omega_i), \dots, a_m(\omega_i), \lambda(\omega_i))$, obtaining a *dataset of examples*, moreover the product space $X = X_1 \times \cdots \times X_m$ forms a *lattice* $(X, \leq)$ where for each $x, y \in X$,

$$x \leq y \Leftrightarrow x_j \leq y_j, \text{ for } j = 1, \dots, m. \qquad (1)$$

We say that the dataset of examples is *consistent* if and only if for each $\omega_i, \omega_h \in \Omega$ it holds $(a_1(\omega_i), \dots, a_m(\omega_i)) = (a_1(\omega_h), \dots, a_m(\omega_h))$ implies $\lambda(\omega_i) = \lambda(\omega_h)$, moreover, it is said to be *monotone consistent* if and only if for each $\omega_i, \omega_h \in \Omega$ it holds $(a_1(\omega_i), \dots, a_m(\omega_i)) \leq (a_1(\omega_h), \dots, a_m(\omega_h))$ implies $\lambda(\omega_i) \leq \lambda(\omega_h)$.

Recall that each attribute $a_j \in \mathcal{A}$ as well as the labelling function λ determines a partition of Ω, whose elements are denoted, respectively, as $\{a_j = x_{j_s}\} = \{\omega_h \in \Omega : a_j(\omega_h) = x_{j_s}\}$, $s = 1, \dots, t_j$, and $\{\lambda = c_q\} = \{\omega_h \in \Omega : \lambda(\omega_h) = c_q\}$, $q = 1, \dots, k$, moreover, the same partitions can be object-wise written, denoting for each $\omega_i \in \Omega$

$$[\omega_i]_{a_j} = \{\omega_h \in \Omega : a_j(\omega_i) = a_j(\omega_h)\}, \qquad (2)$$

$$[\omega_i]_{\lambda} = \{\omega_h \in \Omega : \lambda(\omega_i) = \lambda(\omega_h)\}, \qquad (3)$$

where for each $\omega_h \in [\omega_i]_{a_j}$ we have $[\omega_h]_{a_j} = [\omega_i]_{a_j}$ and, analogously, for each $\omega_h \in [\omega_i]_\lambda$ we have $[\omega_h]_\lambda = [\omega_i]_\lambda$.

2.1. Rank version of conditional Shannon entropy [15]

In [15] the following object-wise writing of conditional Shannon entropy is shown (reported here in our notation).

Proposition 2.1 *Put* $p_s = \frac{|\{a_j = x_{j_s}\}|}{|\Omega|}$ *and* $p_{q,s} = \frac{|\{\lambda = c_q\} \cap \{a_j = x_{j_s}\}|}{|\Omega|}$*:*

$$\begin{aligned} H_S(\lambda|a_j) &= \sum_{s=1}^{t_j} p_s \left(-\sum_{q=1}^{k} \left(\frac{p_{q,s}}{p_s} \right) \log_2 \left(\frac{p_{q,s}}{p_s} \right) \right) \\ &= \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \left(-\log_2 \left(\frac{|[\omega_i]_\lambda \cap [\omega_i]_{a_j}|}{|[\omega_i]_{a_j}|} \right) \right). \end{aligned}$$

In the same paper the authors underline the incapability of conditional Shannon entropy to detect monotonicity of λ w.r.t. a_j. To overcome this obstacle, they go back to the dominance rough set approach (see [12, 13]) introducing the notion of *dominant set* generated, respectively, by a_j and λ. For each $\omega_i \in \Omega$, they define

$$[\omega_i]^{\leq}_{a_j} = \{\omega_h \in \Omega \,:\, a_j(\omega_i) \leq a_j(\omega_h)\}, \quad (4)$$

$$[\omega_i]^{\leq}_{\lambda} = \{\omega_h \in \Omega \,:\, \lambda(\omega_i) \leq \lambda(\omega_h)\}. \quad (5)$$

Then they propose a rank version of Shannon conditional entropy, obtained simply substituting in the object-wise writing the equivalence classes $[\omega_i]_\lambda \cap [\omega_i]_{a_j}$ and $[\omega_i]_{a_j}$ with the corresponding dominant sets, deriving the following definition of the *Shannon rank discrimination measure* (we keep conditional notation for uniformity):

Definition 2.1

$$H^*_S(\lambda|a_j) = \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|}\left(-\log_2\left(\frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}\right)\right).$$

In Definition 2.1, the ratio $\frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}$ is a measure of satisfaction of the *local monotonicity constraint* for a fixed $\omega_i \in \Omega$, quantifying the validity of

$$a_j(\omega_i) \leq a_j(\omega_h) \;\Rightarrow\; \lambda(\omega_i) \leq \lambda(\omega_h),$$

for all $\omega_h \in \Omega$.

It is easy to see that, for a fixed $\omega_i \in \Omega$,

$$\frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|} = 1$$

if and only if $a_j(\omega_i) \leq a_j(\omega_h) \;\Rightarrow\; \lambda(\omega_i) \leq \lambda(\omega_h)$, for all $\omega_h \in \Omega$, since $\frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|} = 1$ if and only if $\{\omega_h \in \Omega \,:\, \lambda(\omega_i) \leq \lambda(\omega_h) \wedge a_j(\omega_i) \leq a_j(\omega_h)\} = \{\omega_h \in \Omega \,:\, a_j(\omega_i) \leq a_j(\omega_h)\}$ and this is true if and only if the local monotonicity constraint is satisfied for ω_i.

In the rest of the paper, to simplify notation, for a fixed $a_j \in \mathcal{A}$ and λ denote:

$$\mathrm{dsr}(\omega_i) = \frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}, \quad (6)$$

$$\mathrm{mindsr}(\omega_i) = \frac{\min\limits_{a_j(\omega_h)=a_j(\omega_i)} |[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}, \quad (7)$$

$$\mathrm{maxdsr}(\omega_i) = \frac{\max\limits_{a_j(\omega_h)=a_j(\omega_i)} |[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}, \quad (8)$$

$$\mathrm{avgdsr}(\omega_i) = \frac{\sum\limits_{a_j(\omega_h)=a_j(\omega_i)} \frac{|[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}|}{|[\omega_i]_{a_j}|}}{|[\omega_i]^{\leq}_{a_j}|}. \quad (9)$$

2.2. New rank discrimination measures

Here, we present the extension of the approach proposed by Hu et al in [15] to other well-known discrimination measures such as Gini measure and Yuan and Shaw measure and we investigate if the obtained functions are proper rank discrimination measures. For more details and proofs, see [23].

In analogy with Definition 2.1, we define the *Gini rank discrimination measure* from the Gini measure by replacing the equivalence classes $[\omega_i]_\lambda \cap [\omega_i]_{a_j}$ and $[\omega_i]_{a_j}$ in its object-wise writing with the corresponding dominant sets. Thus we obtain:

Definition 2.2

$$H^*_G(\lambda|a_j) = \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|}\left(1 - \mathrm{dsr}(\omega_i)\right).$$

Notice that the rank generalization of Gini measure given in Definition 2.2 differs from the one proposed in [31].

The same procedure is applied to the Yuan and Shaw measure of ambiguity [32]. In order to achieve the object-wise writing, we have to notice that in the standard definition of this measure a total order on the cardinalities

$$|\{\lambda = c_q\} \cap \{a_j = x_{j_s}\}|, \quad q = 1, \ldots, k,$$

is assumed for each fixed $s = 1, \ldots, t_j$, therefore, we need to "transport" this ordinal structure to objects in Ω by defining a proper binary relation $\precsim_{\lambda,a_j}$ [23].

To introduce the rank version of Yuan and Shaw measure, a definition of relation $\precsim_{\lambda,a_j}$ taking into account dominant sets is needed.

Definition 2.3 *For each* $\omega_i, \omega_h \in \Omega$

$$\omega_i \precsim'_{\lambda,a_j} \omega_h \quad \textit{iff} \quad \begin{array}{c} [\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j} = [\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j} \\ \textit{or} \\ \left\{\begin{array}{l} a_j(\omega_i) = a_j(\omega_h) \\ [\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j} \neq [\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j} \\ |[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}| \leq |[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}| \end{array}\right. \end{array}.$$

It is easy to show that relation $\precsim'_{\lambda,a_j}$ is a partial preorder on Ω, moreover the symmetric part $\sim'_{\lambda,a_j}$ is an equivalence relation on Ω while the asymmetric part $\prec'_{\lambda,a_j}$ is a partial strict order on $\Omega_{/\sim'_{\lambda,a_j}}$. In particular, to each decreasing $\succ'_{\lambda,a_j}$-chain in $\Omega_{/\sim'_{\lambda,a_j}}$ we can associate an increasing index starting from 1 and so for each ω_i we can define $\rho'(\omega_i) =$ "index of the $\sim'_{\lambda,a_j}$-equivalence class containing ω_i", and the *rank version of Yuan and Shaw measure* is defined as:

Definition 2.4

$$H^*_Y(\lambda|a_j) = \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \cdot \frac{\log_2\left(\mathrm{fin}\left(\frac{\rho'(\omega_i)}{\rho'(\omega_i)-1}\right)\right)}{\mathrm{maxdsr}(\omega_i)},$$

with $\mathrm{fin}(x) = \left\{\begin{array}{ll} x & \textit{if } x < \infty \\ 1 & \textit{otherwise} \end{array}\right.$.

Observing Definition 2.4 one can see that the rank version of Yuan and Shaw measure considers for each object ω_i only the set $[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}$ with the maximum cardinality having $a_j(\omega_h) = a_j(\omega_i)$. This optimistic approach may produce a sort of blindness w.r.t. monotonicity since $\text{maxdsr}(\omega_i)$ could be 1 even if $\text{dsr}(\omega_i)$ is less than 1. Furthermore, the measure takes into account an ordering on the cardinalities of dominant sets (expressed by ρ') which may conflict with the order on the values determining the dominant sets themselves, which is the one we wish to preserve. In detail, in [23] we have shown that H^*_Y is not a good rank discrimination measure, thus we will not take it into account in the rest of the paper.

In the next definition we introduce directly a third measure which is inspired to the functional structure of Definition 2.4 but has a cautious nature and it is called *pessimistic*. The *Pessimistic rank discrimination measure* is defined as:

Definition 2.5

$$H^*_P(\lambda|a_j) = \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|}\left(-\frac{\log_2(\text{mindsr}(\omega_i))}{\text{mindsr}(\omega_i)}\right).$$

We stress that the ratio $\text{mindsr}(\omega_i)$ can be equal to 1 only in the case $\text{dsr}(\omega_i)$ is 1 but it could be less than 1 even in the case the last equality holds.

2.3. Rank discrimination capabilities

It can be proven that the proposed rank discrimination measures satisfy a set of good properties to be used as discrimination measures [22, 23], moreover in [23] the following theorem has been proven.

Theorem 2.1 $H^*_G(\lambda|a_j) = H^*_S(\lambda|a_j) = H^*_P(\lambda|a_j) = 0$ *if and only if* λ *is monotone w.r.t.* a_j*, that is for each* $\omega_i, \omega_h \in \Omega$*,*

$$a_j(\omega_i) \leq a_j(\omega_h) \Rightarrow \lambda(\omega_i) \leq \lambda(\omega_h).$$

3. RDMT(H^*) classifier

In order to make a comparison of the introduced measures with other proposal present in the literature we wrote a binary decision tree classifier in Java relying on the WEKA package [30]. Our proposal is essentially based on REMT classifier [14], differing from it for the use of a rank discrimination measure H^* for splitting instead of the rank mutual information. We call it RDMT(H^*) classifier, where the acronym RDMT stands for Rank Discrimination Measure Tree. RDMT(H^*) is a simple classifier parametrized by the choice of a rank discrimination measure H^* between H^*_G, H^*_S and H^*_P and by other three pre-pruning parameters. No post-pruning is executed on the resulting tree, and missing values are not allowed. As common practice in tree induction [5], the RDMT(H^*) algorithm is completely specified once are known the following parts: *splitting rule*, *stopping rule* and *labelling rule*. The algorithm proceeds recursively applying this three rules, working at each step on a local set of objects Ω_α where $\Omega_0 = \Omega$.

For the splitting rule, since we restrict to binary trees, each attribute a_j must be binarized as it is done in [26] for numeric attributes. In detail, if $X_j = \{x_{j_1}, \ldots, x_{j_{t_j}}\}$, we denote with $a_j^{x_{j_s}}$ the binary attribute defined as

$$a_j^{x_{j_s}}(\omega_i) = \begin{cases} 0 & a_j(\omega_i) \leq x_{j_s} \\ 1 & \text{otherwise} \end{cases}.$$

Now the splitting rule consists simply in finding the binary attribute $a_*^{x_*}$ minimizing $H^*(\lambda|a_j^{x_{j_s}})$, where a_* is the attribute for splitting and x_* is the splitting value.

Then the local object set Ω_α is partitioned into two subsets, according to $a_*(\omega_i) \leq x_*$ or $a_*(\omega_i) > x_*$, and the procedure is repeated on these two subsets.

We stop growing the tree in the case λ is constant on Ω_α, moreover, to avoid overfitting, three pre-pruning parameters determine further stopping conditions. The parameter $measureThreshold$ sets a lower bound for the rank discrimination measure, the parameter $maxDepth$ sets the maximum length of a path from the root to a leaf node and the parameter $percMinSize$ sets the minimum size of the current object set Ω_α, which is computed as $percMinSize \cdot |\Omega|$. Notice that, since H^*_G, H^*_S and H^*_P have different range, the parameter $measureThreshold$ is deeply tied to the chosen measure and so it must be properly estimated.

Once a stopping condition is reached a leaf node is created and is properly labelled [14]. If λ is constant on Ω_α then the constant value is chosen as label, otherwise if λ is not constant, then the median value is taken. In the particular case λ assumes only two values $c_{l_1} < c_{l_2}$, both on the same number of objects of Ω_α, then c_{l_1} is chosen in the case of a left leaf node, while c_{l_2} is chosen in the case of a right leaf node.

It is important to notice that generally, even if the training dataset is monotone consistent, the greedy tree induction with H^*_S, H^*_G and H^*_P does not guarantee a globally monotone classifier.

Hence, it is important to investigate if RDMT(H^*) can assure at least a weaker form of monotonicity. Algorithm REMT is shown to guarantee a weak kind of monotonicity that we call *rule monotonicity* [14]. Let $\mathcal{T} = (N, A)$ be the decision tree generated by the induction procedure, where $N = \{r\} \cup I \cup L$ is the set of nodes (partitioned in the singleton formed by the root r, the set of internal nodes I and the set of leaves L) and A is the set of directed arcs. It is known (see, e.g., [26]) that each path $r \rightsquigarrow l$ with $l \in L$ induces a decision rule R_l, thus we denote with $\mathcal{R}_\mathcal{T}$ the set of decision rules generated by $\mathcal{T}$.

Given $l_1, l_2 \in L$, R_{l_1} and R_{l_2} are *comparable* (see [14]) only in the case they are generated by the same attributes, in this case we say that $R_{l_1} < R_{l_2}$ if and only if attribute values of R_{l_1} are less than R_{l_2}. We simply denote with $\lambda(R_l)$ the label attached to leaf node l. Then we say that $\mathcal{T}$ is *rule monotone* if and only if for each $R_{l_1}, R_{l_2} \in \mathcal{R}_\mathcal{T}$:

$$R_{l_1} < R_{l_2} \ \Rightarrow\ \lambda(R_{l_1}) < \lambda(R_{l_2}). \qquad (10)$$

In [14] it is proven that in the case the dataset is monotone consistent, then algorithm REMT guarantees rule monotonicity. The following proposition states that the same also holds for RDMT(H^*) [23].

Proposition 3.1 *Let $\mathcal{D}$ be a dataset of examples $\{(a_1(\omega_i), \dots, a_m(\omega_i), \lambda(\omega_i)) : i = 1, \dots, n\}$ and $\mathcal{T}$ a binary decision tree built with RDMT(H^*) on $\mathcal{D}$, where $H^* \in \{H_S^*, H_G^*, H_P^*\}$. If $\mathcal{D}$ is monotone consistent then $\mathcal{T}$ is rule monotone.*

4. Experimental analysis

The RDMT(H^*) algorithm for monotone decision tree construction has been compared with other well-known monotone and non-monotone classifiers having a WEKA implementation, on classification tasks involving artificial and real datasets. Each test has been executed performing a stratified 10-folds cross-validation with the same seed (equal to 1) for the pseudo-casual number generator: the WEKA environment guarantees all the folds are equal for each tested classifier.

We used WEKA 3-6-0 since it is the last version providing implementations of the used monotone classifiers. Two non-monotone classifiers have been considered: `J48` which is a Java implementation of C4.5 classifier [26] and `SimpleCart` which is a Java implementation of CART classifier [5]. For what concerns the monotone classifiers, we used `OLM`, `OSDL` and `OCC` which are, respectively, Java implementations of the Ordinal Learning Model [5], the Ordinal Stochastic Dominance Learner [5] and the Ordinal Class Classifier [11] (this last classifier is a monotone meta-classifier for which we used `J48` as basic classifier). Table 1 lists the used classifiers and the related characteristics.

Classifier	Monotone	Globally monotone
RDMT(H^*)	yes	no
C4.5	no	no
CART	no	no
OLM	yes	yes
OSDL	yes	yes
OCC	yes	no

Table 1: Used classifiers

For each test we considered the percentage of Correctly Classified Instances, or CCI for short, the Kappa statistic, or K for short, and the Mean Absolute Error, or MAE for short.

Remark 4.1 *Recall that K ranges in $[-1, 1]$ and is a measure of accuracy corrected for random successes [4]. In detail, a classifier is as much more accurate as K is close to* 1.

In all the following tests we used the default parameter settings for the WEKA implementations of all the considered classifiers, as reported in the WEKA explorer.

Firstly we compared the classifiers on artificial data, producing datasets with an increasing number of monotone attributes. For $k = 1, \dots, 10$, we generated a dataset of 1000 examples on 10 attributes, where a_j is a uniform random variable on $\{1, \dots, 10\}$, $j = 1, \dots, 10$, and the labelling function is defined as $\lambda = \max_{j=1,\dots,k} a_j$. Clearly, for $k = 10$ the corresponding dataset is monotone consistent due to monotonicity of maximum operator.

In order to execute a fair comparison between the three measures H_G^*, H_S^* and H_P^* we set $maxDepth = 100$, $measureThreshold = 0$ and $percMinSize = 0.01$ for all the tests on artificial data. Indeed, setting $measureThreshold > 0$ could favour some measure and penalize the others. Figure 1 displays graphics of CCI, K and MEA for each classifier.

Observing Figure 1 one can see that RDMT(H_G^*) achieves the best results presenting the highest CCI and K together with the lowest MEA for all k except for $k = 9$ in which CART has slightly better results.

Each one of RDMT(H_G^*), RDMT(H_S^*) and RDMT(H_P^*) behaves systematically better than monotone classifiers OLM, OSDL and OCC also for high values of k in which the degree of monotonicity in the dataset tends to be perfect. In particular, one would expect much better accuracy results in case of monotone consistency of the dataset, i.e., for $k = 10$. These three monotone algorithms seem to appreciate the increment of monotonicity in the dataset but their improvement in accuracy is very small.

For what concerns non-monotone classifiers, RDMT(H_G^*) and RDMT(H_S^*) behave always better than C4.5, while RDMT(H_P^*) presents slightly less values of CCI compared to C4.5 for $k = 10$, having anyway a greater value of K and a lesser value of MEA. Taking into account CART we have that it always performs better than RDMT(H_P^*) while it has slightly better results with respect to RDMT(H_S^*) only for $k = 9, 10$, and RDMT(H_G^*) only for $k = 9$, but it has a greater MEA.

Comparing the different rank discrimination measures, it is immediate to verify that for this test RDMT(H_G^*) behaves better than both RDMT(H_S^*) and RDMT(H_P^*), while worse results are achieved by RDMT(H_P^*). It is important to notice that for each measure H^* the performances of RDMT(H^*) tend to degrade for increasing k (as it happens for non-monotone classifiers C4.5 and CART), where

Dataset	#Instances	#Attributes	#Classes
Employee Rejection Acceptance (ERA)	1000	4	4
Employee Selection (ESL)	488	4	9
Lectures Evaluation (LEV)	1000	4	5
Social Workers Decisions (SWD)	1000	10	4
CPU	209	6	10
Breast Cancer	277	9	2
Dermatology	358	34	4
Lymphography	148	18	4

Table 2: Tested datasets

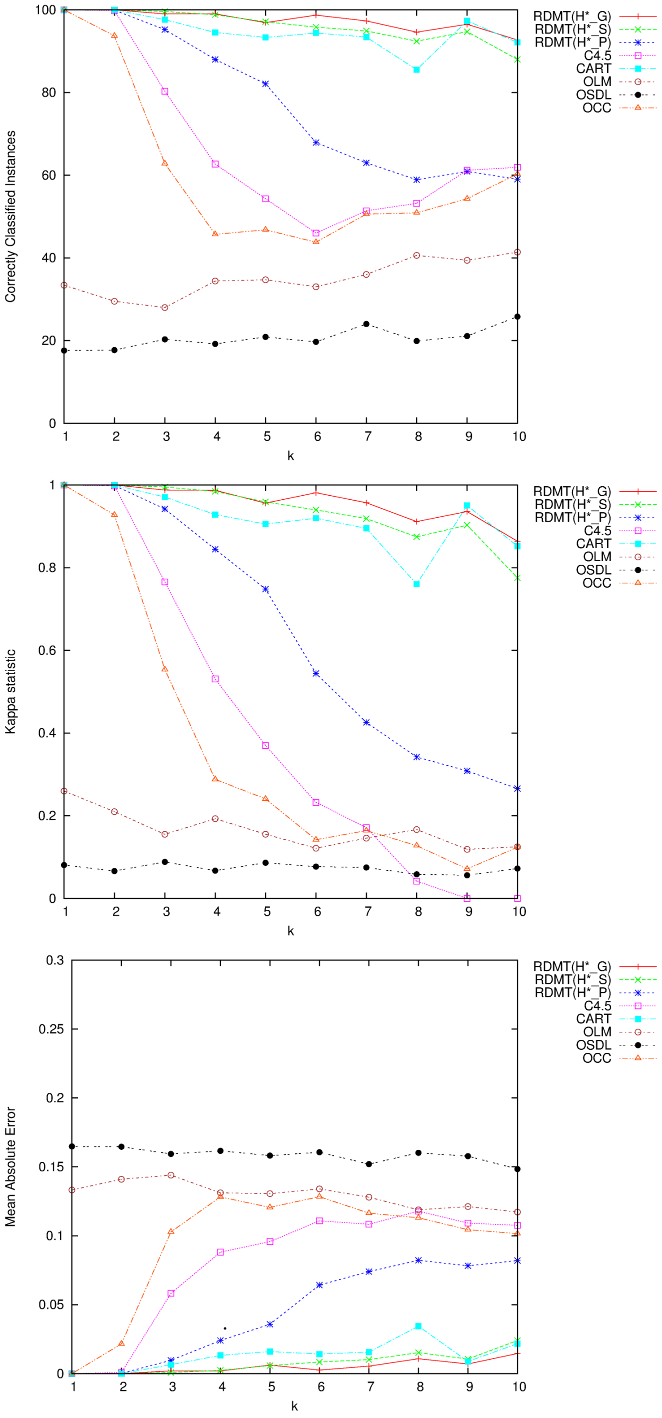

Figure 1: CCI, K and MEA on artificial data

the degrade in performance is much more considerable for H^*_P. We relate the behaviour of measure H^*_P to its pessimistic nature which does not allow to exploit a high number of monotone attributes.

The given results show that RDMT(H^*) outperforms all the other classifiers for k comprised between 3 and 7, that is in presence of remarkable non-monotone noise.

We conclude this section reporting results of tests executed on real datasets taken from UCI [28] and WEKA [30] repositories. Since `OLM` and `OSDL` do not support numeric attributes or examples with missing values, we preprocessed all the datasets, removing all the examples with missing values and discretizing real attributes applying WEKA filter `Discretize`, moreover, integer attributes have been converted to nominal ones by using WEKA filter `NumericToNominal`, both contained in `weka.filters.unsupervised.attribute`.

We collected eight datasets whose description (taking into account the pre-processing phase) is given in Table 2.

After a preliminary estimation study, we set the pre-pruning parameters to $maxDepth = 100$, $percMinSize = 0.01$, while $measureThreshold = 0.01$ for H^*_G, $measureThreshold = 0.05$ for H^*_S and $measureThreshold = 0.1$ for H^*_P in a way to obtain a good compromise between accuracy and tree size.

Table 3 shows results of tests executed on datasets listed in Table 2. Those results essentially highlight a good behaviour of RDMT(H^*) also on real data. In detail, the first four datasets (*ERA*, *ESL*, *LEV* and *SWD*) are monotone datasets taken from WEKA repository while the other four (*CPU*, *Breast Cancer*, *Dermatology* and *Lymphography*) are general purpose datasets.

In the first four datasets, RDMT(H^*) occupies always one of the first three positions in the rank of accuracy results between the six analysed classifiers: it is in first position for *LEV*, in second position for *ERA* and *ESL*, and in third position for *SWD*. Notice that in the three cases in which RDMT(H^*) is not in first position the difference in CCI with the first position in the accuracy rank is always strictly less than 1%. The obtained results are clearly a consequence of the chosen parameter setting and a different choice could further increase the performance.

Dataset		RDMT(H_G^*)	RDMT(H_S^*)	RDMT(H_P^*)	C4.5	CART	OLM	OSDL	OCC
ERA	CCI	26.30%	26.30%	26.30%	26.70%	24.60%	24.40%	23.60%	23.50%
	K	0.1304	0.1304	0.1304	0.1405	0.1140	0.1005	0.0975	0.0875
	MEA	0.1638	0.1638	0.1638	0.1769	0.1783	0.1680	0.1698	0.1854
ESL	CCI	67.62%	63.72%	65.77%	65.98%	63.93%	54.09%	68.23%	60.86%
	K	0.5942	0.5452	0.5702	0.5705	0.5445	0.4269	0.6016	0.5018
	MEA	0.0719	0.0806	0.0760	0.1021	0.1030	0.1020	0.0706	0.1160
LEV	CCI	63.70%	63.70%	63.70%	60.40%	63.30%	46.30%	63.10%	60.20%
	K	0.4772	0.4772	0.4772	0.4320	0.4741	0.2424	0.4665	0.4190
	MEA	0.1452	0.1452	0.1452	0.2025	0.1985	0.2148	0.1476	0.2055
SWD	CCI	58.50%	58.50%	58.30%	56.50%	57.80%	47.10%	58.70%	58.90%
	K	0.3622	0.3622	0.3590	0.3332	0.3582	0.1985	0.3636	0.3661
	MEA	0.2075	0.2075	0.2085	0.2668	0.2738	0.2645	0.2065	0.2575
CPU	CCI	84.21%	83.25%	83.73%	84.68%	85.64%	85.64%	80.38%	82.29%
	K	0.5849	0.5490	0.5866	0.5865	0.6092	0.6075	0.5401	0.5359
	MEA	0.0316	0.0335	0.0325	0.0348	0.0379	0.0285	0.0392	0.0443
Breast Cancer	CCI	68.59%	68.59%	67.50%	74.36%	71.84%	65.34%	53.06%	74.36%
	K	0.1721	0.1785	0.1666	0.2535	0.1541	0.1375	0.1157	0.2535
	MEA	0.3141	0.3141	0.3249	0.3682	0.3737	0.3466	0.4693	0.3682
Dermatology	CCI	89.66%	86.31%	91.89%	93.57%	94.97%	89.38%	12.01%	86.59%
	K	0.8710	0.8281	0.8990	0.9192	0.9370	0.8676	−0.0292	0.8314
	MEA	0.0345	0.0456	0.0270	0.0292	0.0244	0.0354	0.2933	0.0600
Lymphography	CCI	71.62%	75.00%	79.72%	79.72%	76.35%	40.54%	56.75%	82.43%
	K	0.4567	0.5206	0.6178	0.6169	0.5378	0.1680	0.2536	0.6626
	MEA	0.1419	0.1250	0.1014	0.1258	0.1448	0.2973	0.2162	0.1168

Table 3: Results concerning CCI, K and MEA of tests on real datasets

Also in the last four datasets, RDMT(H^*) remains always in the first three positions being second (ex aequo with C4.5) on *Lymphography* and third in all the other three datasets. Notice that in each case the difference in CCI with the first position in the accuracy rank is always strictly less than 6%.

>From previous discussion we can conclude that our classifier, essentially based on a rank discrimination measure, can compete with more sophisticated ones having also a pruning phase, such as C4.5 and CART.

We want to underline that in all the tested real datasets the best performances of RDMT(H^*) are always obtained by H_G^* or H_P^* (in four of the eight cases there is an ex aequo with H_S^*) thus these new two measures appear to behave generally better than H_S^*.

5. Conclusion

In this paper, we presented a rank generalization of Gini discrimination measure and Yuan and Shaw discrimination measure, moreover we introduced directly a third function inspired to the functional structure of the second generalized measure.

We also presented a binary tree classifier RDMT(H^*) parametrized by a rank discrimination measure H^* and other three pre-pruning parameters. RDMT(H^*) has been implemented in Java using the WEKA package and it has been tested on artificial and real datasets, comparing it with other well-known monotone and non-monotone classifiers also implemented in WEKA. This classifier assures a weak form of monotonicity on the resulting tree, namely *rule monotonicity*, in the case the dataset is monotone consistent. Our analysis shows our classifier can exploit the eventual monotonicity of the dataset: it can compete with non-monotone classifiers in accuracy and, moreover, it is much more robust to non-monotone noise than purely monotone classifiers. Thus an empirical proof of effectiveness of the proposed rank discrimination measures is given.

In future work, we plan the fuzzification of the proposed rank discrimination measures in a way to deal with fuzzy decision tree classifiers [17, 32].

References

[1] N. Abu-Halaweh and R. Harrison. Practical fuzzy decision trees. In *Proc. of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, pages 211 – 216, Nashville, (USA), March 2009.

[2] A. Ben-David. Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, 19:29–43, 1995.

[3] A. Ben-David, L. Sterling, and Y. Pao. Learning and classification of monotonic ordinal concepts. *Computational Intelligence*, 5(1):45–49, 1989.

[4] A. Ben-David, L. Sterling, and T. Tran. Adding monotonicity to learning algorithms may impair their accuracy. *Expert Systems with Applications*, 36(3, Part 2):6627–6634, 2009.

[5] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees.* Chapman and Hall/CRC, Boca Raton, 1984.

[6] K. Cao-Van and B. De Baets. Consistent representation of rankings. In H. de Swart, E. Orlowska, G. Schmidt, and M. Roubens, editors, *Theory and Applications of Relational Structures as Knowledge Instruments*, volume 2929

of *Lecture Notes in Computer Science*, pages 1966–1967. Springer Berlin / Heidelberg, 2003.

[7] K. Cao-Van and B. De Baets. Growing decision trees in an ordinal setting. *International Journal of Intelligent Systems*, 18(7):733–750, 2003.

[8] K. Crockett, Z. Bandar, and D. Mclean. On the optimization of t-norm parameters within fuzzy decision trees. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1–6, July 2007.

[9] L.-C. Dong, D. Wang, and X.-Z. Wang. An improved sample selection algorithm in fuzzy decision tree induction. In *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernectics*, pages 629–634, San Antonio, TX (USA), October 2009.

[10] A. Feelders. Monotone Relabeling in Ordinal Classification. In *IEEE International Conference on Data Mining 2010 (ICDM 2010)*, pages 803–808, 2010.

[11] E. Frank and M. Hall. A simple approach to ordinal classification. In L. D. Raedt and P. Flach, editors, *12th European Conference on Machine Learning - ECML 2001*, volume 2167 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2001.

[12] S. Greco, B. Matarazzo, and R. Slowinski. Rough approximation by dominance relations. *International Journal of Intelligent Systems*, 17(2):153–171, 2002.

[13] S. Greco, B. Matarazzo, and R. Slowinski. Rough sets methodology for sorting problems in presence of multiple attributes and criteria. *European Journal of Operational Research*, 138(2):247–259, 2002.

[14] Q. Hu, X. Che, L. Zhang, D. Zhang, M. Guo, and D. Yu. Rank entropy based decision trees for monotonic classification. *Knowledge and Data Engineering, IEEE Transactions on*, page (in publishing), 2011.

[15] Q. Hu, M. Guo, D. Yu, and J. Liu. Information entropy for ordinal classification. *SCIENCE CHINA Information Sciences*, 53:1188–1200, 2010.

[16] Q. Hu, W. Pan, L. Zhang, D. Zhang, Y. Song, M. Guo, and D. Yu. Feature selection for monotonic classification. *Fuzzy Systems, IEEE Transactions on*, 20(1):69–81, 2012.

[17] E. Hüllermeier and S. Vanderlooy. Why fuzzy decision trees are good rankers. *Fuzzy Systems, IEEE Transactions on*, 17(6):1233–1244, 2009.

[18] S. Lievens and B. De Baets. Supervised ranking in the WEKA environment. *Information Sciences*, 180(24):4763–4771, 2010.

[19] X. Liu and W. Pedrycz. The development of fuzzy decision trees in the framework of axiomatic fuzzy set logic. *Applied Soft Computing*, 7:325–342, 2007.

[20] C. Marsala and B. Bouchon-Meunier. Quality of measures for attribute selection in fuzzy decision trees. In *Proceedings of the International Conference on Fuzzy Systems (WCCI-2010)*, Barcelona, Spain, July 2010.

[21] C. Marsala, B. Bouchon-Meunier, and A. Ramer. Hierarchical model for discrimination measures. In *Proc. of the IFSA'99 World Congress*, pages 339–343, Taiwan, 1999.

[22] C. Marsala and D. Petturiti. Hierarchical model for rank discrimination measures. In *Proc. of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2013)*, page (to appear), 2013.

[23] C. Marsala and D. Petturiti. Rank discrimination measures for monotone decision tree induction. *Information Sciences*, (submitted).

[24] W. Pedrycz and Z. Sosnowski. Genetically optimized fuzzy decision trees. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(3):633–641, June 2005.

[25] R. Potharst and A. Feelders. Classification trees for problems with monotonicity constraints. *SIGKDD Explorations Newsletter*, 4(1):1–10, 2002.

[26] J. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[27] A. Tehrani, W. Cheng, and E. Hüllermeier. Choquistic Regression: Generalizing Logistic Regression using the Choquet Integral. In *Proc. of the 7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2011)*, pages 868–875, 2011.

[28] UCI. UC Irvine Machine Learning Repository. `http://archive.ics.uci.edu/ml/`.

[29] X. Wang and C. Borgelt. Information measures in fuzzy decision trees. In *Proc. of the IEEE International Conference on Fuzzy Systems*, volume 1, pages 85–90, July 2004.

[30] WEKA. Machine Learning Group at University of Waikato. `http://www.cs.waikato.ac.nz/ml/weka/`.

[31] F. Xia, W. Zhang, and F. L. Y. Yang. Ranking with decision tree. *Knowledge and Information Systems*, 17(3):381–395, 2008.

[32] Y. Yuan and M. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69(2):125–139, 1995.