Expert Systems With Applications 152 (2020) 113375

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A weighted information-gain measure for ordinal classification trees

Gonen Singer[a,*], Roee Anuar[b], Irad Ben-Gal[b]

[a] Bar-Ilan University, Faculty of Engineering, Ramat-Gan, 52900, Israel
[b] Tel-Aviv University, Department of Industrial Engineering, Tel-Aviv, 39040, Israel

ARTICLE INFO

Article history:
Received 7 August 2019
Revised 23 February 2020
Accepted 10 March 2020
Available online 13 March 2020

Keywords:
Information-gain
Decision trees
Classification tree
Weighted entropy
C4.5
Ordinal classification

ABSTRACT

This paper proposes an ordinal decision-tree model, which applies a new weighted information-gain ratio (WIGR) measure for selecting the classifying attributes in the tree. The proposed measure utilizes a weighted entropy function that is defined proportionally to the value deviation of different classes and thus reflects the consequences of the magnitude of potential classification errors. The WIGR can be used to select the classifying attributes in decision trees in a manner that reduces risks. The proposed ordinal decision tree is found effective for classification problems in which the class variable exhibits some form of ordinal ordering, and where dependencies between the attributes and the class value can be non-monotonic. In a series of experiments based on publicly-known datasets, it is shown that the proposed ordinal decision tree outperforms its non-ordinal counterparts that utilize traditional entropy measures. The proposed model can be used as a part of an expert system for ordinal classification applications, such as health-state monitoring, portfolio investments classification and performance evaluation of service systems.

1. Introduction

Ordinal classification problems are often treated as multi-class classification problems, in which the target class exhibits some form of ordinal ordering. These problems commonly address real-world applications for which expert-systems and machine-learning algorithm were developed, such as automatic classification of severity of diseases (Nabi et al., 2019), portfolio investment by expected returns (Altuntas and Dereli, 2015) or performance prediction of queueing systems (Senderovich et al., 2015). In these problems, it is important to take into account the value deviation among the different classes, since the magnitude of potential classification error could results in critical consequences (Gaudette et al., 2009), such as the detection of the level of congestive heart failure (Masetic and Subasi, 2016) or prediction of load level in emergency services (Senderovich et al., 2015; Mouroo et al., 2017; Sanit-in and Saikaew, 2019). Most of the ordinal classification methods in the literature assume monotonic behavior, according to which the class values follow a monotonicity constraint with respect to the classifying attributes (Marsala and Petturiti, 2013; Ben-David, 1995; Ben-David et al., 1989; Zhu et al., 2017; Verbeke et al., 2017). Such a constraint implies that the influence of an attribute on the class, either increases or decreases throughout the attribute range. Thus, these methods assume that the classifying attributes also follow some natural ordering, which is not necessarily true in many real cases. Accordingly, these approaches cannot deal with categorical attributes, nor with attributes that affect the class variable non-monotonically in some range sections. In fact, non-monotonic dependencies between explanatory attributes and target class are very common in many ordinal problems, for example, on medical application of congestive heart failure identification the ordinal attribute “blood pressure” may have a non-monotonic effect on the level of congestive heart failure (e.g., extreme blood pressure values, either high or low, may lead to high level of congestive heart failure while under “regular” blood pressure the level of congestive heart failure may be low). Another example can be prediction of load level in service system, in which the ordinal attribute “weather forecast” may have a non-monotonic effect on the load of a service system (e.g., extreme weather conditions, either hot or stormy, may lead to lower loads while under “regular” conditions the load may be higher). Specifically, the monotonicity assumption hinders the use of decision trees, in which categorical attributes are commonly used. Decision-tree algorithms are known to be efficient and interpretable models for representing classification problems. In most of these algorithms, including ID3, C4.5, and Random Forest (see Shiju and Remya, 2014; Fernández-Delgado et al., 2014), Shannon’s entropy (Shannon, 1948) plays a fundamental role in attribute

* Corresponding author.
E-mail addresses: gonen.singer@biu.ac.il (G. Singer), roee.anuar@gmail.com (R. Anuar), bengal@tauex.tau.ac.il (I. Ben-Gal).

https://doi.org/10.1016/j.eswa.2020.113375
0957-4174/

selection. In the context of classification problems, Shannon's entropy is often used to measure the quantity of information that a classifying (explaining) attribute provides about the class variable. Note, however, that Shannon's entropy does not reflect well ordinal class variables, since the entropy solely depends on the distribution of possible outcomes rather than the outcome values and their associated effects. This implies, for example, that a distribution of four symbols in a class variable that represents an expected return on investment and given by the vector (30%, 10%, 20%, 40%), results in the same entropy measure as the distribution (10%, 30%, 40%, 20%), although the ordinality of the two vectors and therefore their potential effects on the expected return and the associated risks might be significantly different, as seen in the running example presented below. Several researchers have extended entropy-based algorithms to take into account the ordinal behavior of the class, but most of them assume monotonicity not only of the class variable but also of the explaining attributes. Ben-David (1995) introduced a non-monotonicity index defined as the ratio between the actual number of non-monotonic branch-pairs of a decision tree and the maximum number of pairs that are non-monotonic with respect to each other in the same tree. Potharst and Bioch (2000) proposed an algorithm for repairing non-monotonic decision trees for multi-attribute classification problems with several linearly ordered classes. Other researchers proposed tree-induction algorithms to deal with the monotonicity of data (Cao-Van and Baets, 2003; Potharst and Feelders, 2002). Hu et al. (2010) generalized Shannon's entropy to address *crisp ordinal classification* as well as *fuzzy ordinal classification* and proposed indices to evaluate the degree of monotonicity between attributes and the decision in the context of ordinal classification. Hu et al. (2012) designed a decision-tree algorithm based on a new measure of attribute quality called *rank mutual information* (REMT) to build a monotonically-consistent decision tree when the training samples are monotonically consistent. Qian et al. (2015) proposed a so-called "fusing monotonic decision trees" (FREMT) algorithm which combines decision trees with an ensemble-learning technique. The method obtained an improved classification performance. Cardoso and Sousa (2011) surveyed different approaches for measuring the success of an ordinal classification, including measures that are applied in the numerical study of the current research.

Despite the large body of research on different ordinal classifiers, there is no comprehensive study that compares their performance. Moreover, it is not clear whether these ordinal models outperform their non-ordinal sibling classifiers. In fact, Ben-David et al. (2009) showed that the ordinal classifiers were statistically indistinguishable from their non-ordinal counterparts since the monotonicity assumption led to high levels of non-monotonic noise data that resulted in a poor classification accuracy.

In this research we propose an extended ordinal decision-tree, which is based for simplicity reasons on the well-known C4.5 algorithm but does not depend on any monotonicity constraint of the classifying attributes. We use a risk-based information gain ratio as an attribute selection criterion for ordinal classification tasks. The concept of using a risk-based entropy measure originates from the portfolio management literature (e.g., see Philippatos and Wilson, 1972; Dionisio et al., 2006; Ormos and Zibriczky, 2014; and Mahmoud and Naouib, 2017). We conduct a numerical study, which is based on well-known ordinal datasets with a high level of non-monotonic noisy data. Similarly to Ben-David et al. (2009), the performance of the proposed algorithm is compared to that of its non-ordinal counterpart, namely the conventional C4.5 algorithm itself. The results show that the proposed tree-generation algorithm outperforms the non-ordinal algorithm for most of the datasets. Thus, the contribution of this work is three-fold. First, it extends the use of Information-Gain measure, which is based on the classic definition of entropy as measure of uncertainty, to deal with ordinal targets taking into account the value deviations of different target classes. Note that Information-Gain is a centric measure in many experts and intelligent systems including ones that are based on Decision Trees, Bayesian models, Reinforcement Learning and more. Second, the proposed measure is practically used in this study for selecting branching (classifying) attributes in decision trees over both ordinal and non-ordinal categorical variables, and shown to yield a very different outcomes from the conventional non-ordinal approaches. Finally, the study presents a systematic experimental analysis framework and demonstrates the superiority of the proposed algorithm relative to its non-ordinal version over many publically-known ordinal datasets that are related to intelligent systems.

This research is novel both in its goal of developing an ordinal decision-tree model, which is based on a new *weighted information-gain ratio* (WIGR) measure. This measure utilizes a weighted entropy function that is defined proportionally to the value of different target classes, and thus reflects consequences of the magnitude of potential classification errors. In comparison to conventional information-gain measures, the proposed WIGR can be used to better select the classifying attributes in a decision tree in a manner that reduces risks in ordinal cases. Furthermore, the proposed ordinal decision-tree does not depend on any monotonicity constraint of the classifying attributes. Thus, it can be used as part of an expert and intelligent systems for ordinal classification problems, in which the magnitude of the classification error can be critical, and non-monotonic dependencies between the explanatory attributes and the target class are very common. Such systems can be developed to support smart application with ordinal targets, such as the identification disease's severity, classification of portfolio investments by expected returns and performance prediction of service systems.

The rest of the paper is organized as follows. Section 2 introduces the concept of using entropy as a risk measure, presents the challenge in using an entropy measure for ordinal classification and proposes an extended information-gain measure. The numerical experiments are then presented in Section 3. Finally, the conclusions and future work directions are discussed in Section 4.

2. Methodology

2.1. Risk-based entropy

Several studies have used the degree of entropy uncertainty as a risk measure in portfolio management. Philippatos and Wilson (1972) presented the case wherein two securities have the same entropy, but different levels of risk. They were among the first researchers who applied entropy to the study of portfolio selection. The authors emphasized the difference between the entropy and the variance, claiming that while the entropy depends on the number and distribution of states in a probability distribution (known as the second property of Shannon and Weaver (1949)), the variance depends on the state values. They found that the entropy provided more general results and sometimes outperformed the standard deviation when measuring risk, since the entropy is a nonparametric measure. Dionisio et al. (2006), Ormos and Zibriczky (2014) and Mahmoud and Naouib (2017) showed that the entropy is sensitive to diversification and uses information on the probability distribution and as result is efficient for measuring risk in portfolio management cases. As such, in general terms it can measure uncertainty better than the variance. Nawrocki and Harding (1986) extended the entropy measure and proposed a weighted entropy as a measure of investment risk, to deal with a case of two different securities holding the same entropy values but different risk levels.

Table 1
Revenue return probabilities of two different investment portfolios with different expected returns but with the same entropy.

Investments		State i					
		Hard Loss (40% return)	Loss (80% return)	Medium Profit (120% return)	High Profit (180% return)	Entropy	Expected Return
Investment #1	p_i	0.3	0.1	0.2	0.4	1.85	116%
Investment #2	p_i	0.1	0.3	0.4	0.2	1.85	112%

Following on from the above literature, this research proposes a new, state-value, weighted-information gain-ratio which is an extended measure of the conventional information-gain ratio (used by many tree-generation algorithms including the C4.5 which is analyzed in this work). The proposed measure relies on a risk-based entropy, which is better suited to loss-oriented and risk-averse tasks than the variance or conventional entropy measures. The latter do not take into consideration the ordinal properties of the class variable when selecting a branching attribute in a decision tree. In other words, when classifying an ordinal target, the classic information metrics do not prefer attributes that partition the data into bins in which the values are more densely distributed in adjacent bins. In contrast, as seen next, the proposed entropy measure takes into consideration the classification expected costs and rewards when selecting a branching attribute in a decision tree.

2.2. Entropy measure of ordinal variables

Shannon's entropy is a mathematical property that measures the randomness and the uncertainty about the outcome of a random variable. The lower the entropy, the more predictable the outcome of the random variable is. This conventionally-used entropy is given by

$$H = -K\sum_{i=1}^{n} p_i \log(p_i), \tag{1}$$

where p_i is the probability mass function of the ith outcome of the class variable and K is considered as a constant that normalizes the information units according to the used logarithm base, where $K = 1$ and logarithm to the base 2 are often used by the conventional entropy that is then measured in bits.

Note that the conventional entropy does not consider the actual values of the random variable, nor the ordinal form associated with it. The entropy treats each outcome as a unique probabilistic event. Thus, different outcomes that have the same probability contribute equally to overall entropy measure, regardless of their potential effect on the expected value. In particular, the entropy ignores the potential costs or rewards that are associated with various outcomes in a risk-management case, such as the case of a portfolio management decision making.

Let us illustrate the challenge of using a conventional entropy measure when considering a running example of a portfolio investment. Table 1 presents two investments, #1 and #2, that have a similar probability distribution, yet with different associated risks. Both investments yield the same entropy value of 1.85 bits, although most investors will prefer investment #1 over investment #2 since the probability for a "High Profit" is higher (40% vs. 20%) resulting in a higher expected return (1.16 vs. 1.12). Later however, we show that when using the proposed weighted information gain, there is an advantage for an investor to select Investment #2 from the perspective of prediction, decision-making and the weighted risk.

Awared about the above-mentioned limitation, Nawrocki and Harding (1986) proposed a weighted entropy measure that can be used to reflect the risk level of each outcome in addition to its probability distribution, i.e.,

$$H = -\sum_{i=1}^{n} k_i p_i \log(p_i). \tag{2}$$

The authors considered two forms of the state-dependent weights k_i that can be also applied for illustration purpose to the protfolio-management example in Table 1. Namely, while comparing investment #1 versus investment #2, one can set the weights to be proportional to either i) the squared deviations from the mean; or ii) the absolute deviations from the mean. Thus, unlike Shannon's entropy, which has a fixed K=1 assuming all bits of information are uniformly distributed over all the realizations, Nawrocki and Harding suggested using weights k_i with different component-wise values. We extend this notion by taking into account the associated monetary values and by generalizing the assumption that the expected loss is uniformly distributed over all realizations. As seen next, we integrate this approach with an information-gain ratio measure for selecting a branching attribute in a conventional decision tree. Note that this approach has some similarities with, yet is significantly different from other entropy measures, such as the Tsallis entropy and the Rényi entropy, which contain additional parameters. These parameters can be used to make the measure more or less sensitive to the shape of the probability distribution; however, it is not related to the potential gain or risk values of the outputs, as proposed here.

2.3. Weighted information gain for selecting branching attributes in a decision tree

Decision trees are one of the most popular and commonly used types of classification algorithm. A variety of decision-tree algorithms are used for data mining purposes, such as ID3, C4.5 and CART (Breiman, Friedman, Olshen, & Stone, 1984; Quinlan, 1986, 1993; Singer & Golan, 2019; Singer, Golan, Rabin, & Kleper, 2019), while ensembles of these models, such as random forests and rotational forests, have been found to yield good and robust classification results (see Byon et. al, 2010; Shiju and Remya, 2014; Fernández-Delgado et al., 2014; Jin and Deng, 2015, Zhao et al., 2016, Bacher and Ben-Gal, 2017).

In this research, we explore publicly-known classification databases. Each dataset contains an ordinal categorical class (target) variable, as well as possible explanatory attributes that can take different numbers of values, ranging from just a few to a very large number of values. Some of the datasets are also exposed to monotonic noise over the attributes. In the experiment, we benchmark the proposed Weighted- Information Gain-Ratio (WIGR) against the conventional Information-Gain Ratio (IGR). In particular, the WIGR is used to select a branching attribute at each node of the decision tree, in a similar procedure to which is applied in conventional decision trees. The proposed WIGR can be considered as an extension of the information-gain measure, in the sense that it adjusts to the case of an ordinal target and takes into account the value deviation among different target classes. Let D be an m-row dataset with $J + 1$ columns, columns 1, ..., J representing the attributes, where each attribute A_j holds one of α_j possible values (not necessarily ordinal), so that $dom(A_j) = \{A_j^r, r = 1, \ldots, \alpha_j\}$, where *dom* denotes the domain. Column $J + 1$ represents the class

variable C with n possible classes, so that $dom(C) = \{C_i, i = 1, \ldots, n\}$ is an ordered numeric set with ordinal property of the class variable, such that $V(C_i) \leq V(C_l), \forall i < l$, where $V(C_i)$ is the reward value or the cost value associated with class C_i reflecting the ordinality of the class variable. In the example presented in Table 1, $V(C_i)$ reflect the return percentages of an investment, where 40% < 80% < 120% < 180%. Furthermore, when relying on these values one can calculate the value deviation among different target classes, as explained above, and reflect the critical consequences of the magnitude of the classification error. Thus, for the investments in Table 1, classifying a 'Hard Loss' investment (with a 40% return) as a 'High Profit' investment (with a 180% return) reflects a higher error than classifying a 'Medium Profit' investment (with a 120% return) as a 'High Profit' investment. Recall that Shannon's entropy is a function of the random variable probabilities only, and does not consider the dispersion of the outcomes and their associated rewards. Note, however, that in a real usecases not only that each event has a certain occurance probability, but also it is associated with a different reward or cost value.

Accordingly, in the first step of the proposed approach, we define a *weighted entropy* (WH) measure that allocates weights (often unequal) to different classes, namely

$$\mathrm{WH}(D) = -\sum_{i=1}^{n} k_i(D)\, p_i \log p_i, \;\; 0 \leq k_i(D) \leq 1 \sum_{i=1}^{n} k_i(D) = 1, \qquad (3)$$

where p_i is the probability for a record to be associated with class C_i over a consider dataset D, while k_i is the corresponding non-negative weight of class C_i for D . Note that in the context of a decision tree the weighted entropy is calculated in each node with respect to its associated subset $D \subseteq \mathrm{D}$. In the root node, prior to any partitioning, the distribution of the classes is calculated over the entire dataset D. However, in descendant nodes that follow a selection criterion of branching attribute(s), the distribution of the classes is calculated over subsets $D \subseteq \mathrm{D}$ following the dataset partitioning by the tree. In this research, we propose a simple calculation to allocate weights to the different classes according to their reward values and dispertion with respect to the class mode. This measure implies that an attribute with a denser distribution around the mode value, i.e., the value of the class with the highest probability (the highest frequency in D), obtains a smaller WH value, since it represents a lower risk. Specifically, we propose that the values of the weights k_i will be derived by the difference between the value of each class $V(C_i)$ and the value of the most probable class, in the dataset D, as follows:

$$k_i(D) = \frac{\left|V\left(C_i\right) - V\left(C^{mode}(D)\right)\right|^{\propto}}{\sum_{i=1}^{n}\left|V\left(C_i\right) - V\left(C^{mode}(D)\right)\right|^{\propto}}, \qquad (4)$$

where $V(C^{mode}(D))$ is the value of class mode thus the value of the most probable class in D. Thus, $k_i(D)$, represents the absolute deviation of the value of the i-th class from the value of the class mode in the considered dataset, divided by the sum of all absolute deviated values over all possible classes in the dataset, such that the sum of the weights is equal to one and $\propto$ ($\propto > 0$), is a normalization factor that smooths the weights' distribution over the different classes. The larger is $\propto$, the larger are the weights of the classes that are distant from the most-probable class (the class mode), and the smaller are the weights of classes that are closer to the most-probable class. In other words, this parameter controls the sensitivity with respect to cases that are distant from the class mode. For example, assume one has to choose between the distribution $P = (0.51,\ 0.49,\ 0)$ derived from dataset D_1 and the distribution $Q = (0.95,\ 0,\ 0.05)$ derived from dataset D_2. Assuming $V(C_i) = i,\ i = 1,\ 2,\ 3$ for $\propto = 1$, one will prefer Q over P (as $\mathrm{WH}(D_2) = 0.144 < \mathrm{WH}(D_1) = 0.168$), while for $\propto = 2$ the decision alters (as now $\mathrm{WH}(D_2) = 0.173 > \mathrm{WH}(D_1) = 0.101$), since the weight (thus the "penalty") is higher for classes that are distant from the class mode. Note that the weight's values, k_i, of the classes are calculated by Eq. (4), which is derived from the values of each class $V(C_i)$ and the selection of the most-probable class, C^{mode}. Such a selection is straightforward if there is one class that has the highest-frequency in the dataset, but it is not clear if two or more classes share the same highest-frequency in the dataset. In such a case, any selection of the class mode will change the weights' values and as a result the weighted entropy, unless the class values are fully symmetric. Since for given a specific selection of C^{mode} the weights of the classes are fixed, a rule of the thumb is to allocate the C^{mode} to a class that can increase the prediction preferences by the decision maker. For example, a risk-averse decision maker will allocate the C^{mode} such that the weighted entropy is lower for attributes with higher predictability power of classes that can result in significant losses, since he wants to avoid such cases. Similarly, a risk-seeker will look for attributes with better predictability of classes that can lead to extensive gains, as he wants to identify such opportunities ahead. Allocation of the C^{mode} over the median class, even if it is not the most probable class, could be the strategy in case that the decision-maker has no preferable prediction class. Another approach that could be followed, is to calculate the weighted entropy over all the different allocations of the C^{mode}, while averaging them to a single weighted measure. Doing so, the weighted measure represents the average marginal contribution over all possible allocations in a manner similar to the calculation of the known shapley value. Part of this research is left for future studies and addressed in the Conclusions Section.

Now, following a conventional decision-tree procedure, suppose one aims to partition the dataset D represented in some node by a branching attribute A_j having α_j distinct values. The partition is defined by the different branching branches from the node that represent all the records in dataset D, where each branch is defined by one out of α_j distinct values of the branching attribute A_j. The conditional entropy of the partitioning attribute A_j is given by

$$WH_j(D) = \sum_{r=1}^{a_j} \frac{\left|D_j^r\right|}{|D|} \times WH\left(D_j^r\right) \qquad (5)$$

where $WH_j(D)$ is the weighted-entropy measure of a possible partitioning over the attribute A_j, and D_j^r is the subset of records of D for which $A_j = A_j^r$, thus $D_j^r = \{D|A_j = A_j^r\}$. The Weighted Information Gain of attribute A_j over D, denoted by WIG_j, indicates how much information is gained by branching D over the values of A_j, i.e.,

$$\mathrm{WIG}_j(D) = \; WH(D) - WH_j(D) \qquad (6)$$

Note that this measure can be negative for specific branches of A_j.

The associated partition entropy of A_j is given by

$$H_j(D) = -\sum_{r=1}^{a_j} \frac{\left|D_j^r\right|}{|D|} \times \log_2\left(\frac{\left|D_j^r\right|}{|D|}\right). \qquad (7)$$

Since the attribute A_j is not necessarily ordinal, we use here a conventional information measure in Eq. (1). Finally, by dividing Eq. (6) by Eq. (7), the normalized Weighted Information Gain Ratio (WIGR), when attribute A_j is selected as the branching attribute over dataset D, is calculated as follows

$$\mathrm{WIGR_j}(D) = \; \frac{\mathrm{WIG}_j(D)}{H_j(D)} \qquad (8)$$

The attribute with the highest (non-negative) normalized weighted information gain ratio is selected by the proposed procedure as the branching attribute of the node that represents the

Algorithm 1: Pseudo code for tree growing

Input: D, **where** D= dataset of classified instances
Output: *Ordinal Decision Tree*
Require: $D \neq \emptyset$, $num\_attributes>0$
Procedure Build tree(D)

```
Tree={}
if D is "pure then
    STOP;
End if
MaxWIGR ← 0
jMAX ← null
For all Attribute j belong to D do
        WIGR_j ← WIGR_j(D)
        if WIGR_j > MaxWIGR then
            MaxWIGR ← WIGR_j
        end if
        jMax = arg max_{j∈D} (WIGR_j)
end for
if MaxWIGR > 0 then
    Tree = create a decision node that test jMax in the root;
    D_v ← induced sub datasets from D based on jMax
    for all D_v do
        Tree_v ← Build tree(D_v)
        Attach Tree_v to the corresponding branch of tree
    End for
end if
return tree
```

Fig. 1. The Meta-code of the proposed WIGR Algorithm for tree growing.

population D. If all attributes have a zero or a negative value, the tree-construction procedure is stopped. For comparison purpose, let us denote the normalized classic information gain ratio, when attribute A_j is selected as the branching attribute over dataset D, by $IGR_j(D)$. This measure is based on the conventional entropy measure in Eq. (2) where all the weights are equal, $k_i = 1$, $\forall i$ (Quinlan,1986).

$$IGR_j(D) = \frac{H(D) - H(D|A_j)}{H(A_j)} \tag{9}$$

The proposed tree-construction algorithm is relatively simple. Instead of using the conventional information gain ratio as a branching criterion over all the possible branching attributes, it uses the normalized weighted information gain ratio in Eq. (8) to select the branching attribute. The WIGR algorithm is described in Fig. 1. As seen in the pseudo code, the tree building is a recursive procedure. Starting with the entire dataset D. Then in in each step, the procedure finds the attribute with the highest weighted information gain ratio. Once such an attribute is found with a positive weighted information gain, the procedure call itself, using the relevant partition as the new dataset, while splitting on each value of the selected attribute.

Table 2 presents both the conventional Shannon's Entropy, based on Eq. (1), as well as the proposed weighted entropy, based on Eqs. (3) (4) with $\propto = 2$, assuming $V(C_i) = i$, for the two investments given in Table 1. Note that in this example, both investments obtain the same conventional information gain. Nonetheless, investment #2 is preferable over investment #1 once the associated risks and gains are considered, as measured by the weighted-information gain scores and as explained below. Such a selection is in oppose to the one which is based solely on the expected return that places investment #1 as preferable.

Table 2
The entropy and the weighted entropy measures of the two investments portfolios shown in Table 1.

	H (Investment)	WH (Investment)
Investment #1	1.85	0.46
Investment #2	1.85	0.39

Let us continue further with the running example from Table 1 to describe the proposed WIGR measure and its use for selecting the classifying (branching) attributes for the decision-tree construction. Table 3 shows the dataset associated with Investment #2, which was presented in Table 1, where in 40% of the investments result in loss (eight rows with the value "1"), 40% of the investments result in medium profit (eight rows with the value "2") and 20% of the investments result in High Profit (four rows with the value "3"). Accordingly, the third column represents the class variable of each record as follows: "1" represents the consolidated "Loss" and "Hard Loss" cases; "2" represents "Medium Profit" cases, and "3" represents "High Profit" cases. The branching attribute has to be selected among columns A_1, A_2 that represent the classify-

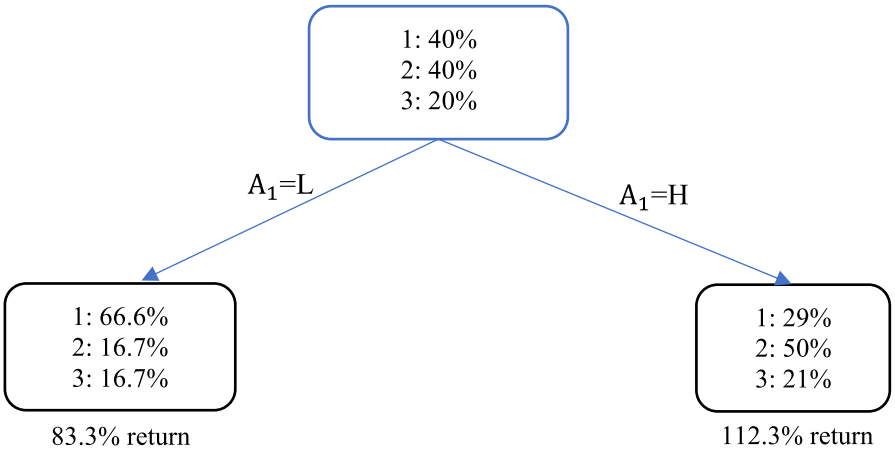

Fig. 2. A one-level C4.5 decision tree for the dataset *D*, given in Table 3, based on the conventional information-gain ratio that is applied to Investment #2 in Table 1.

Table 3
An illustrative dataset with 20 instances based on the values in Table 1.

Attribute 1	Attribute 2	Class (Investment #2)
L	H	1 - Loss
H	H	1 - Loss
L	H	1 - Loss
H	H	2 - Medium Profit
H	H	2 - Medium Profit
H	L	2 - Medium Profit
H	H	2 - Medium Profit
H	L	2 - Medium Profit
L	H	1 - Loss
H	H	2 - Medium Profit
H	H	3 - High Profit
L	H	1 - Loss
H	H	1 - Loss
H	L	3 - High Profit
H	H	3 - High Profit
H	L	1 - Loss
L	L	3 - High Profit
H	L	2 - Medium Profit
H	H	1 - Loss
L	H	2 - Medium Profit

ing (explanatory) attributes. The $WIGR_j(D)$ represents the potential weighted-information gain when selecting A_j as the branching attribute over dataset D, while consolidating for simplicity reasons the two loss states in Table 1 into one loss state with a respective return of 50%. This new measure can be benchmarked against the conventional information-gain measure, $IGR_j(D)$, which is often used as a branching criterion by many decision tree algorithms, including the classical C4.5.

At this stage one has to select the branching attribute for the 20-instances dataset D that consists of the class variable and two input attributes, A_1 and A_2. According to the classical information-gain ratio in Eq. (9), $IGR_1(D) = 0.12$ if A_1 is selected as the branching attribute, and $IGR_2(D) = 0.09$, if A_2 is selected as the branching attribute. Thus, relying the classical information gain measure (e.g., see Maimon and Rokach, 2005), one should select attribute A_1 over A_2 as the branching attribute, resulting in the one-level C4.5 decision tree shown in Fig. 2. Note, however, that when using the weighted-information gain-ratio (WIGR) by Eq. (8), attribute A_1 obtains a value of $WIGR_1(D) = 0.015$, if it is selected as the branching attribute, while attribute A_2 obtains a corresponding $WIGR_2(D) = 0.04$. Thus, when using the WIGR measure, A_2 should be selected as the branching attribute, resulting in the weighted one-level C4.5 decision tree shown in Fig. 3. As seen, Fig. 3 represents a less-risky decision tree compared to the tree in Fig. 2. More specifically, in Fig. 2, both leaves represent a gain distribution which is more likely to generate a lower return, since when A_1=L, there is a probability of 67% for a loss with an expected return of 83.3%, and when A_1=H, there is a probability of 29% for a loss with an expected return of 112.3%. Obviously, a rational decision maker will select the A_1=H branch, resulting in the higher return value. In Fig. 3, when A_2=L, there is a probability of 17% for a loss with an expected return of 127.9% and when A_2=H, there is a probability of 50% for a loss with an expected return of 93.4%. Here a rational decision maker will select the first branch with the highest expected return over all the other options. Thus, based on the decision maker selections A_2 is preferable.

In the next section we apply the proposed tree-generation algorithm based on the WIGR measure to twelve publicly-known databases to classify the ordinal targets. The purpose of these experiments is to assess the performance of the proposed algorithm in comparison to the conventional, non-ordinal decision-tree algorithms. In the considered examples, there are various attributes with different numbers of values, including both ordinal and categorical class variables.

3. Experimental results

To test the new ordinal classification approach, we conducted a classification test over twelve publicly known datasets, which were previously tested by Gutierrez et al. (2016). For simplicity of exposition, we mapped the 10-class problems into 3-class problems,so the probabilities of the different new classes will be roughly uniformly distributed, labeling classes 1-3 as 'Low', classes 4-7 as 'Medium' and classes 8-10 as 'High', such that each run is performed over three ordinal classes. We benchmarked the ordinal-weighted C4.5 decision tree, which is based on the proposed WIGR measure with a normalization factor of $\propto = 1$ in Eq. 4, for selecting the branching attributes against the conventional and popular C4.5 decision tree version. Nevertheless, note that the WIGR measure could also be implemented with other trees. To evaluate the best classification model, we used a 20-fold 'leave-one-out' test, as suggested by Gutierrez et al. (2016). The classification accuracy is evaluated and averaged over the 20 folds, each time leaving out one of the sub-samples and using it as a test case. The recall, the precision and the errors metrics are calculated for each class. Table 4 presents the average recall over the entire testing sets obtained from the 20 folds test, using both the conventional and the weighted-ordinal C4.5 classifier. As can be seen from the table, the proposed weighted-ordinal classifier tends to yield a better recall for the boundary classes (i.e., those labeled as 'Low' and 'High'),

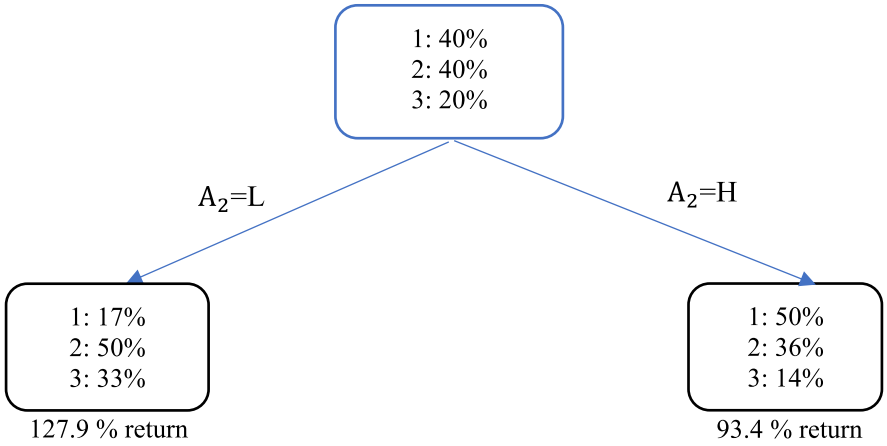

Fig. 3. A one-level C4.5 decision tree for the dataset *D*, given in Table 3, based on the proposed weighted information-gain ratio that is applied to Investment #2 in Table 1.

Table 4
Average recall over the tested datasets, comparing both the conventional C4.5 and the proposed ordinal weighted classifier for each of the classes. Higher values are bolded.

Dataset	Regular C4.5			Weighted-ordinal C4.5		
	Class=Low	Class=Med	Class=High	Class=Low	Class=Med	Class=High
Abalone	56.9%	**54.0%**	62.4%	**76.8%**	41.2%	**71.8%**
Bank1	84.6%	67.9%	65.6%	**91.9%**	**68.4%**	**82.1%**
Bank2	61.4%	**51.8%**	54.9%	**82.0%**	49.1%	**78.9%**
Calhousing	71.0%	**53.5%**	69.0%	**78.4%**	49.1%	**80.0%**
Census1	73.8%	**67.1%**	69.6%	**87.4%**	51.0%	**82.8%**
Census2	73.8%	**59.1%**	70.4%	**88.8%**	48.1%	**83.7%**
Computer1	79.8%	**71.2%**	79.8%	**90.2%**	60.5%	**89.5%**
Computer2	79.1%	**69.6%**	80.1%	**90.4%**	60.1%	**89.5%**
Housing	81.6%	67.9%	72.9%	**84.5%**	**69.7%**	**73.2%**
Machine	66.9%	**60.4%**	75.6%	**75.6%**	58.5%	**79.5%**
Pyrim	44.6%	**56.3%**	6.7%	**56.8%**	39.2%	**22.5%**
Stock	91.5%	**84.5%**	85.4%	**94.6%**	77.3%	**94.7%**

Table 5
Average precision over the tested datasets, comparing both the conventional C4.5 and the weighted ordinal classifier for each of the classes. Higher values are bolded.

Dataset	Regular C4.5			Weighted-ordinal C4.5		
	Class=Low	Class=Med	Class=High	Class=Low	Class=Med	Class=High
Abalone	65.1%	51.8%	**58.3%**	**69.7%**	**57.9%**	56.2%
Bank1	85.9%	64.8%	70.2%	**87.4%**	**78.8%**	**75.3%**
Bank2	63.1%	49.5%	57.0%	**69.3%**	**73.3%**	**63.9%**
Calhousing	63.2%	60.1%	**68.4%**	**67.1%**	**69.2%**	66.4%
Census1	**68.7%**	64.3%	**81.1%**	68.4%	**75.1%**	73.0%
Census2	66.2%	61.8%	**75.2%**	**69.1%**	**78.2%**	68.8%
Computer1	**79.6%**	71.3%	**80.6%**	74.7%	**81.9%**	80.2%
Computer2	**79.2%**	70.2%	79.8%	74.0%	**81.0%**	**80.8%**
Housing	77.2%	68.8%	77.9%	**77.3%**	**70.9%**	**80.3%**
Machine	**68.0%**	63.2%	77.5%	66.1%	**67.5%**	**83.3%**
Pyrim	**43.4%**	41.9%	34.6%	39.3%	**45.5%**	**47.4%**
Stock	**91.8%**	83.5%	**87.9%**	88.4%	**90.8%**	84.6%

while the regular C4.5 tree yield a better recall for the central class (labeled as 'Med').

Table 5 presents the average precision over the entire testing sets obtained from the 20-folds test, for both types of classifiers. In 24 out of the 36 comparisons (67%), the weighted-ordinal classifier outperformed the regular C4.5 algorithm. For the central 'Med' class, the weighted-ordinal classifier obtained superior precision over the entire collection of datasets. Combining the results from Tables 4 and 5, in 50 out of the 72 comparisons (69.4%), the weighted-ordinal classifier outperformed the popular C4.5 method in both the recall and the precision, while, in 14 out of the 36 comparisons, the weighted-ordinal classifier outperformed the popular C4.5 method in both the recall and the precision simultaneously. Conversely, there were no cases in which the conventional C4.5 classifier outperformed the weighted-ordinal classifier in both measures.

To derive a measure of performance that takes into account both the precision and the recall simultaneously, we calculate, per each dataset, the *F-Score*:

$$F - Score = \frac{2\,Precision\,Recall}{Recall + Precision}, \tag{10}$$

which can be is interpreted as a weighted average of the precision and recall, ranging between zero to one as its highest score. Table 6 presents the *F-Score* for each of the datasets. As seen, the weighted-ordinal C4.5 classifier outperformed the conventional

Table 6
Average F-scores over the tested datasets, comparing both the conventional C4.5 and the weighted ordinal classifier for each of the classes. Higher values are bolded.

Dataset	Regular C4.5			Weighted-ordinal C4.5		
	Class=Low	Class=Med	Class=High	Class=Low	Class=Med	Class=High
Abalone	57.5%	**52.0%**	60.2%	**73.0%**	48.0%	**63.0%**
Bank1	84.9%	66.0%	67.5%	**89.4%**	**72.7%**	**77.9%**
Bank2	62.0%	50.3%	55.5%	**74.8%**	**58.1%**	**70.4%**
Calhousing	66.7%	56.4%	68.6%	**72.1%**	**57.2%**	**72.5%**
Census1	71.0%	**65.6%**	74.7%	**76.6%**	60.5%	**77.5%**
Census2	69.7%	**60.2%**	72.5%	**77.6%**	59.1%	**75.4%**
Computer1	79.5%	**71.1%**	80.0%	**81.3%**	69.1%	**84.4%**
Computer2	79.0%	**69.7%**	79.7%	**81.3%**	68.7%	**84.8%**
Housing	79.0%	68.0%	74.9%	**80.5%**	**69.9%**	**76.3%**
Machine	68.2%	59.6%	77.8%	**69.2%**	**60.8%**	**80.4%**
Pyrim	40.7%	**53.0%**	40.6%	**50.9%**	45.5%	**48.8%**
Stock	**91.5%**	**83.8%**	86.3%	91.3%	83.4%	**89.3%**

one in 28 out of the 36 comparisons (78%). Note also that the weighted classifier outperformed its conventional counterpart in all but one of the boundary classes ('Low' and 'High'). For the 'Med' classes, the performance of the two classifiers is approximately equal, emphasizing again the contribution of the WIGR in ordinal cases.

To test whether the overall difference in the performance metrics between the two classifiers is significant, we conducted paired t-tests. For each dataset, a paired t-test was carried out using 20 pairs of results, each of which was based on the average performance over all classes for a given cross-validation. Table 7 summarizes the *t-test* results for each of the datasets for the precision, the recall and the *F-Score*. *p*-values lower than 0.05 are in bold font and those lower than 0.1 are in *italic* font. For all datasets, the average values of all the three-performance metrics – the recall, the precision and the *F-score* – are higher when applying the proposed weighted-ordinal C4.5 classifier. This difference is found to be significant at a 90% level for the recall in 100% of the datasets, for the *F-score* in 92% of the datasets and for the precision measure in 75% of the datasets.

Another alternative benchmark approach was discussed in Cardoso and Sousa (2011). In particular, the authors surveyed different approaches for measuring the success of an ordinal classification over a group of datasets. A proposed approach the mentioned is to calculate the mean square error (MSE), defined as:

$$MSE = \frac{1}{N}\sum_{n\in N}\left(V(C_n) - V\left(\hat{C}_n\right)\right)^2 \tag{11}$$

where N is the number of samples, C_n is the real class of sample n, $\hat{C}_n$ is the estimated class of sample n and $V(.)$ corresponds to a number assigned to each class, in our case, $V(C_i) = V(\hat{C}_n) = i,\ i = 1, 2, 3$. A lower MSE score represents a smaller classification error. The MSE measure was calculated for each cross-validation over each dataset. Table 8 presents the average MSE for each dataset, for the two classification methods, along with the *p*-value of the corresponding *t-test*. Once again, this paired t-test rely on the 20 pairs of cross-validations results. As seen from the table, the proposed ordinal decision tree performed better in all but one of the datasets, 9 of which have a significantly better MSE score.

Another interesting performance measure is Kendall's correlation coefficient, τ_b, which is a measure of concordance or the ordinal association between two measured quantities. It is defined as:

$$\tau_b = \frac{B - Z}{\sqrt{B + Z + e_t}\sqrt{B + Z + e_b}}, \tag{12}$$

where B refers to the number of concordant pairs in the classification, i.e., pairs in which the relative ordering of the "real" classes $V(C_i)$ and $V(C_j)$ is the same as the relative ordering of the classified classes $V(\hat{C}_i)$ and $V(\hat{C}_j)$, and Z refers to the number of discordant pairs in the classification, i.e., pairs in which the relative ordering of the "real" classes is opposite to the relative ordering of the classified classes.

The parameter e_t refers to the number of samples that are tied on the true class and e_b refers to the number of samples that are tied on the estimated class. Pairs that hold the same real and estimated classes are ignored. τ_b ranges from -1 to 1. The higher the performance measure, the better the classification performance. We calculated the Kendall's correlation score for each cross-validation of each dataset. Table 9 presents the average correlation score for each dataset and classification method, along with the *p*-value of a *t-test*. As before, the *t-test* was based on the 20 paired scores from the cross-validations. As seen from the table, the proposed ordinal decision tree showed superior performance over all datasets, with a significant *p*-value.

As suggested in the above experiments, the weighted information gain may replace the classic information gain as the decision criterion for selecting the most suitable branching attribute in an ordinal decision tree.

In the above series of experiments based on known datasets, the comparative analysis showed that the proposed weighted-ordinal decision tree significantly outperforms its non-ordinal counterpart. To evaluate the effect of increasing the number of symbols on the model performance, we repeated the 3-classes experiment study, with different mapping into 5-class problems, such that the probabilities of the new classes are uniformly distributed, labeling classes 1-2 as 'Very Low', classes 3-4 as 'Low', classes 5-6 as 'Medium', classes 7-8 as 'High' and classes 9-10 as 'Very High'. Thus, each run is performed over five ordinal classes. Tables 10–12 present the average recall, precision and F-scores of the weighted-ordinal decision tree vs. the conventional C4.5 decision trees over all the twelve datasets. Similar to the 3-classes results, the weighted-ordinal C4.5 classifier yield better recall, precision and F-score for most of classes in all the studied datasets except of the Abalone dataset. Considering the results from Tables 10 and 11, in 107 out of the 120 benchmark comparisons (i.e., in 89.2% compared to 69.4% in the 3-classes problems), the weighted-ordinal classifier outperformed the C4.5 trees both in the recall and in the precision outcomes. Moreover, in 49 out of the 60 comparisons, the weighted-ordinal classifier outperformed the popular C4.5 method in both the recall and the precision simultaneously. As seen from Table 12, the weighted-ordinal C4.5 classifier outperformed the conventional tree in 54 out of the 60 comparisons (90% of the cases compared to 78% in the 3-calss problems).

Table 13 summarizes the *t-test* results for each of the datasets for the precision, the recall and the *F-Score* measures. *p*-values

Table 7
Paired t-test results for each of the datasets for the precision, the recall and the F-Scores of the conventional C4.5 vs. the proposed weighted-ordinal C4.5. *p*-values lower than 0.05 are bolded and those lower than 0.1 are in *italic* font.

T-Test on the Recall Measure	Regular C4.5	Weighted-ordinal C4.5	*p*-value
abalone10	57.8%	**63.3%**	**0.0048**
bank1-10	72.7%	**80.8%**	**0.0000**
bank2-10	56.0%	**70.0%**	**0.0000**
calhousing-10	64.5%	**69.2%**	**0.0000**
census1-10	70.1%	**73.7%**	**0.0000**
census2-10	67.7%	**73.5%**	**0.0000**
computer1-10	76.9%	**80.1%**	**0.0013**
computer2-10	76.3%	**80.0%**	**0.0003**
housing10	74.1%	**75.8%**	**0.0007**
machine10	67.6%	*71.2%*	*0.0668*
pyrim10	35.8%	**39.5%**	**0.0223**
stock10	87.1%	**88.9%**	**0.0017**
T-Test on the Precision	**Regular C4.5**	**Weighted-ordinal C4.5**	***p*-Value**
abalone10	58.4%	**61.3%**	**0.0240**
bank1-10	73.6%	**80.5%**	**0.0000**
bank2-10	56.6%	**68.8%**	**0.0000**
calhousing-10	63.9%	**67.6%**	**0.0000**
census1-10	71.4%	*72.2%*	*0.0898*
census2-10	67.8%	**72.0%**	**0.0000**
computer1-10	77.1%	**78.9%**	**0.0246**
computer2-10	76.4%	**78.6%**	**0.0185**
housing10	74.6%	**76.2%**	**0.0384**
machine10	68.4%	**72.3%**	**0.0403**
pyrim10	39.2%	41.7%	0.2263
stock10	87.7%	87.9%	0.3643
T-Test on F-Score Measure	**Regular C4.5**	**Weighted-ordinal C4.5**	***p*-Value**
abalone10	56.5%	**61.3%**	**0.0203**
bank1-10	72.8%	**80.0%**	**0.0000**
bank2-10	55.9%	**67.8%**	**0.0000**
calhousing-10	63.9%	**67.3%**	**0.0000**
census1-10	70.4%	*71.5%*	*0.0564*
census2-10	67.5%	**70.7%**	**0.0002**
computer1-10	76.9%	*78.3%*	*0.0612*
computer2-10	76.1%	**78.3%**	**0.0156**
housing10	74.0%	**75.6%**	**0.0061**
machine10	68.0%	*70.1%*	*0.0804*
pyrim10	48.9%	50.4%	0.2811
stock10	87.2%	*88.0%*	*0.0702*

Table 8
Paired t-test results for each of the datasets for the MSE of the conventional C4.5 vs. the proposed weighted-ordinal C4.5. *p*-values lower than 0.05 are bolded and those lower than 0.1 are in *italic* font.

T-Test on MSE	Regular C4.5	Weighted-ordinal C4.5	*p*-value
abalone10	0.5021053	0.4917	0.2278
bank1-10	0.2778947	**0.2042**	**6.427E-07**
bank2-10	0.5585965	**0.4421**	**0.0004**
calhousing-10	0.4722807	**0.4358**	**0.0003**
census1-10	0.3627068	**0.3474**	**0.0365**
census2-10	0.3921053	**0.3739**	**0.01617**
computer1-10	0.2521053	*0.2332*	*0.05144*
computer2-10	0.2568421	**0.2286**	**0.0057**
housing10	0.3012281	**0.2788**	**0.0244**
machine10	0.3680702	0.3418	0.1426
pyrim10	0.9873684	0.9937	0.4572
stock10	0.1328947	**0.123**	**0.0393**

Table 9
Paired t-test results for each of the datasets for the Kendall's T_b of the conventional C4.5 vs. the proposed weighted-ordinal C4.5. *p*-values lower than 0.05 are bolded.

T-Test on Kendall Tau	Regular C4.5	Weighted-ordinal C4.5	*p*-value
abalone10	0.3756957	**0.4463804**	**0.0025**
bank1-10	0.5758866	**0.7033707**	**3.293E-08**
bank2-10	0.3325643	**0.5264548**	**2.147E-08**
calhousing-10	0.4437126	**0.5194949**	**7.415E-09**
census1-10	0.5232702	**0.593204**	**8.039E-08**
census2-10	0.494754	**0.5888644**	**1.797E-08**
computer1-10	0.6344298	**0.6920542**	**0.0005592**
computer2-10	0.6263166	**0.6917236**	**1.613E-05**
housing10	0.5909572	**0.6209705**	**0.0004587**
machine10	0.5023906	**0.5624576**	**0.0429549**
pyrim10	0.0359967	**0.1391175**	**0.0094361**
stock10	0.7920192	**0.819621**	**0.0011681**

lower than 0.05 are in bold font and those lower than 0.1 are in *italic* font. For all datasets except one, the average values of all the three-performance metrics – the recall, the precision and the *F-score* – are higher when applying the proposed weighted-ordinal C4.5 classifier. This difference is found to be significant at a 90% confidence level for the recall in 83% of the datasets, for the precision in 75% of the datasets and for the *F-score* measures in 83% of the datasets.

Fine tuning the ordinal decision tree, including the normalization factor $\propto$ in Eq. (4), was found to be useful also in cases when the ordinal weighted C4.5 was compared against other state of the art approaches, such as the popular Random Forest model. For example, Tables 14–16 compare the average recall, precision and F-scores of the weighted C4.5 and the Random Forest for two

Table 10
Average recall values of both the conventional C4.5 classifier and the weighted ordinal classifier for each of the tested classes. Outperforming values are bolded.

Dataset	Regular C4.5 classifier				
	Class=Very Low	Class=Low	Class=Medium	Class=High	Class=Very High
Abalone	**87.89%**	**34.18%**	17.74%	46.03%	33.79%
Bank1	75.79%	52.11%	36.84%	36.84%	46.32%
Bank2	60.53%	30.92%	23.16%	24.44%	43.61%
Calhousing	70.00%	32.28%	29.30%	37.02%	59.65%
Census1	72.22%	39.04%	36.84%	38.39%	64.24%
Census2	71.71%	39.34%	34.61%	42.50%	59.74%
Computer1	77.63%	50.53%	45.79%	49.47%	78.95%
Computer2	74.29%	47.98%	46.95%	48.90%	71.71%
Housing	86.23%	**28.86%**	42.63%	27.19%	78.77%
Machine	75.61%	17.89%	42.46%	19.65%	**82.98%**
Pyrim	73.68%	**23.16%**	1.05%	0.00%	2.11%
Stock	90.57%	**61.14%**	68.73%	70.83%	86.05%
Dataset	Weighted-Ordinal C4.5 classifier				
	Class=Very Low	Class=Low	Class=Medium	Class=High	Class=Very High
Abalone	39.26%	25.34%	**21.29%**	**52.63%**	**36.76%**
Bank1	**87.37%**	**57.37%**	**51.58%**	**68.95%**	**56.84%**
Bank2	***80.59%***	**38.49%**	**41.40%**	**48.12%**	**69.92%**
Calhousing	**78.77%**	**41.40%**	**37.54%**	**46.49%**	**70.35%**
Census1	**82.60%**	**55.56%**	**44.81%**	**53.41%**	**74.46%**
Census2	**88.29%**	**50.92%**	**50.53%**	**51.84%**	**75.53%**
Computer1	**88.95%**	**60.00%**	**59.74%**	**62.63%**	**85.79%**
Computer2	**87.25%**	**54.45%**	**61.68%**	**60.75%**	**86.18%**
Housing	**86.67%**	25.09%	**53.86%**	**29.82%**	**83.60%**
Machine	**76.84%**	**22.46%**	**45.79%**	**33.68%**	80.00%
Pyrim	**75.79%**	21.58%	**2.11%**	**6.84%**	**9.47%**
Stock	**95.31%**	51.40%	**71.40%**	**75.18%**	**86.97%**

Table 11
Average precision values of both the conventional C4.5 classifier and the weighted ordinal classifier for each of the tested classes. Outperforming values are bolded.

Dataset	Regular C4.5 classifier				
	Class=Very Low	Class=Low	Class=Medium	Class=High	Class=Very High
Abalone	**58.25%**	**32.61%**	**31.02%**	**37.14%**	**53.48%**
Bank1	80.00%	51.56%	35.71%	31.67%	54.66%
Bank2	53.64%	30.62%	27.16%	27.31%	39.46%
Calhousing	50.38%	37.55%	39.20%	40.11%	55.19%
Census1	56.52%	42.31%	39.84%	40.92%	69.28%
Census2	56.13%	39.97%	40.15%	45.75%	63.06%
Computer1	70.74%	54.24%	48.07%	52.66%	73.17%
Computer2	69.25%	52.09%	49.56%	45.60%	72.51%
Housing	59.58%	34.63%	40.20%	50.90%	70.05%
Machine	48.05%	38.06%	38.29%	40.88%	60.72%
Pyrim	19.75%	19.13%	**100.00%**	0.00%	44.44%
Stock	**84.67%**	69.08%	61.38%	73.71%	89.22%
Dataset	Weighted-ordinal C4.5 cassifier				
	Class=Very Low	Class=Low	Class=Medium	Class=High	Class=Very High
Abalone	40.64%	28.87%	26.70%	34.92%	43.18%
Bank1	**82.18%**	**67.70%**	**56.98%**	**49.06%**	**72.97%**
Bank2	**59.47%**	**57.64%**	**53.39%**	**56.14%**	**51.52%**
Calhousing	**57.27%**	**49.79%**	**50.95%**	**53.54%**	**59.23%**
Census1	**61.35%**	**57.40%**	**60.20%**	**57.89%**	**73.89%**
Census2	**59.96%**	**59.08%**	**61.64%**	**66.11%**	**71.13%**
Computer1	**74.29%**	**69.30%**	**63.94%**	**68.79%**	**78.55%**
Computer2	**71.00%**	**64.35%**	**68.46%**	**67.23%**	**77.06%**
Housing	**60.87%**	**42.94%**	**41.24%**	**56.48%**	**72.20%**
Machine	**50.29%**	**40.76%**	**39.13%**	**55.01%**	**70.26%**
Pyrim	**22.40%**	**19.25%**	21.05%	**36.11%**	**46.15%**
Stock	76.95%	**70.77%**	**65.51%**	**74.10%**	**93.45%**

Table 12
Average F-scores values of both the conventional C4.5 classifier and the weighted ordinal classifier for each of the tested classes. Outperforming values are bolded.

Dataset	Regular C4.5 classifier				
	Class=Very Low	Class=Low	Class=Medium	Class=High	Class=Very High
Abalone	**70.07%**	**33.38%**	22.57%	41.11%	**41.41%**
Bank1	77.84%	51.83%	36.27%	34.06%	50.14%
Bank2	56.88%	30.77%	25.00%	25.79%	41.43%
Calhousing	58.59%	34.72%	33.53%	38.50%	57.34%
Census1	63.41%	40.61%	38.28%	39.62%	66.67%
Census2	62.97%	39.66%	37.17%	44.07%	61.35%
Computer1	74.03%	52.32%	46.90%	51.02%	75.95%
Computer2	71.68%	49.95%	48.22%	47.20%	72.11%
Housing	70.47%	31.48%	41.38%	35.45%	74.15%
Machine	58.76%	24.34%	40.27%	26.54%	70.13%
Pyrim	31.15%	**20.95%**	2.08%	0.00%	4.02%
Stock	**87.52%**	**64.87%**	64.85%	72.24%	87.61%
Dataset	Weighted-ordinal C4.5 classifier				
	Class=Very Low	Class=Low	Class=Medium	Class=High	Class=Very High
Abalone	39.94%	26.99%	**23.69%**	**41.98%**	39.72%
Bank1	**84.69%**	**62.11%**	**54.14%**	**57.33%**	**63.91%**
Bank2	**68.44%**	**46.15%**	**46.64%**	**51.82%**	**59.33%**
Calhousing	**66.32%**	**45.21%**	**43.23%**	**49.77%**	**64.31%**
Census1	**70.40%**	**56.46%**	**51.38%**	**55.56%**	**74.17%**
Census2	**71.42%**	**54.70%**	**55.53%**	**58.11%**	**73.26%**
Computer1	**80.96%**	**64.32%**	**61.77%**	**65.56%**	**82.01%**
Computer2	**78.29%**	**58.99%**	**64.89%**	**63.82%**	**81.37%**
Housing	**71.52%**	**31.67%**	**46.71%**	**39.04%**	**77.48%**
Machine	**60.79%**	**28.96%**	**42.20%**	**41.78%**	**74.82%**
Pyrim	**34.57%**	20.35%	**3.83%**	**11.50%**	**15.72%**
Stock	85.15%	59.55%	**68.33%**	**74.64%**	**90.10%**

Table 13
Paired t-test results for each of the datasets for the precision, the recall and the F-Scores of the conventional C4.5 vs. the weighted-ordinal C4.5. *p*-values lower than 0.05 are bolded and those lower than 0.1 are in *italic* font.

T-Test on Recall Measures	Regular C4.5	Weighted-ordinal C4.5	*p*-Value
abalone10	**36.61%**	29.21%	**0**
bank1-10	41.32%	**53.68%**	**0**
bank2-10	30.44%	**46.42%**	**0**
calhousing-10	38.04%	**45.76%**	**0**
census1-10	41.79%	**51.81%**	**0**
census2-10	41.32%	**52.85%**	**0**
computer1-10	50.39%	**59.52%**	**0**
computer2-10	48.30%	**58.39%**	**0**
housing10	43.95%	**46.51%**	**0**
machine10	39.77%	**43.13%**	**0.0009**
pyrim10	16.67%	**19.30%**	**0.0065**
stock10	62.89%	63.38%	0.1564
T-Test on Precision Measures	Regular C4.5	Weighted-ordinal C4.5	*p*-Value
abalone10	**42.95%**	37.23%	**0**
bank1-10	51.12%	**67.16%**	**0**
bank2-10	36.07%	**56.66%**	**0**
calhousing-10	44.90%	**54.64%**	**0**
census1-10	60.24%	**71.50%**	**0**
census2-10	49.22%	**63.82%**	**0**
computer1-10	77.10%	**78.90%**	**0**
computer2-10	58.16%	**70.07%**	**0**
housing10	52.72%	**55.87%**	**0.0005**
machine10	48.22%	**53.66%**	**0.0004**
pyrim10	22.63%	24.16%	0.2638
stock10	76.26%	76.42%	0.3998
T-Test on F-Score Measures	Regular C4.5	Weighted-ordinal C4.5	*p*-Value
abalone10	**34.59%**	26.84%	**0.0203**
bank1-10	41.67%	**53.28%**	**0**
bank2-10	29.72%	**45.14%**	**0**
calhousing-10	36.96%	**44.66%**	**0**
census1-10	41.38%	**51.22%**	**0**
census2-10	40.79%	**52.04%**	**0**
computer1-10	49.91%	**59.02%**	**0**
computer2-10	48.04%	**57.84%**	**0**
housing10	42.91%	*43.83%*	*0.1000*
machine10	40.23%	**43.05%**	**0.0109**
pyrim10	17.02%	**21.24%**	**0.0049**
stock10	62.74%	62.85%	0.4217

Table 14
Average recall of the proposed weighted-ordinal C4.5. vs. the Random Forest model for each of the classes in two datasets.

Dataset	Random Forest			Weighted-ordinal C4.5		
	Class=Low	Class=Med	Class=High	Class=Low	Class=Med	Class=High
Bank2	59.9%	**71.9%**	54.4%	**82.0%**	49.1%	**78.9%**
Census1	70.6%	**70.8%**	69.9%	**87.4%**	51.0%	**82.8%**

Table 15
Average precision of the proposed weighted-ordinal C4.5. vs. the Random Forest model for each of the classes in two datasets.

Dataset	Random Forest			Weighted-ordinal C4.5		
	Class=Low	Class=Med	Class=High	Class=Low	Class=Med	Class=High
Bank2	**75.3%**	54.1%	**72.0%**	69.3%	**73.3%**	63.9%
Census1	**71.2%**	63.0%	**84.0%**	68.4%	**75.1%**	73.0%

Table 16
Average F-scores of the proposed weighted-ordinal C4.5. vs. the Random Forest model for each of the classes in two datasets.

Dataset	Random Forest			Weighted-ordinal C4.5		
	Class=Low	Class=Med	Class=High	Class=Low	Class=Med	Class=High
Bank2	66.4%	**61.7%**	61.6%	**74.8%**	58.1%	**70.4%**
Census1	70.8%	**66.5%**	76.1%	**76.6%**	60.5%	**77.5%**

Table 17
Paired t-test results for each of the datasets for the precision, the recall and the F-Scores of the random forest vs. the proposed weighted-ordinal C4.5. *p*-values lower than 0.05 are bolded and those lower than 0.1 are shaded.

T-Test on the Recall Measure	Random Forest	Weighted-ordinal C4.5	*p*-Value
bank2-10	**62.1%**	**70.0%**	**3.11783E-09**
census1-10	**70.4%**	**73.7%**	**2.2582E-05**
T-Test on the Precision Measure	Random Forest	Weighted-ordinal C4.5	*p*-Value
bank2-10	*67.2%*	*68.8%*	*0.075479116*
census1-10	72.7%	72.2%	0.210558305
T-Test on the F-Scores Measure	Random Forest	Weighted-ordinal C4.5	*p*-Value
bank2-10	**63.2%**	**67.8%**	**2.99274E-05**
census1-10	71.1%	71.5%	0.293193973

datasets, namely bank2-10 and census1-10, that were large enough to support the Random Forest learning. Similar to the above results, the weighted-ordinal C4.5 classifier often yield better recall and F-score for the boundary classes (i.e., those labeled as 'Low' and 'High') in comparison to the Random Forest model, while the Random Forest model often resulted in a better recall for the central class (labeled as 'Med'). This observation indicates, once again, that for ordinal cases, where the boundary classes have high importance or gain, the ordinal weighted model can be a good learning model to use.

Table 17 summarizes the *t-test* results of the two datasets for the precision, the recall and the *F-Score*. *p*-values lower than 0.05 are in bold font and those lower than 0.1 are in *italic* font. For the two considered datasets, the average values of the recall and the *F-Score* are higher when applying the proposed weighted-ordinal C4.5 classifier and in one of the datasets for the precision metric as well. Among the cases in which the weighted-ordinal C4.5 outperformed the Random Forest model, the difference is found to be significant at a 90% level in 80% of cases (4 out of 5 cases).

4. Conclusions

This paper addresses an ordinal multi-class classification problem. In particular, it proposes a weighted-ordinal gain ratio (WIGR) that can be used for selecting the branching attribute in associated decision trees. The proposed WIGR is based on an information theoretic measure, adjusted for the case where the target is ordinal and often categorical. The classifying attributes, on the other hand, do not have to follow any natural ordering, as opposed to other known ordinal-classification algorithms. The mo-

tivation for using the WIGR stems from the fact that the conventional information-gain measure takes into consideration only the dispersion (or frequencies) of the classes in each node, while ignoring the state values of the classes and their potential effects on gain and risk factors. Moreover, known ordinal classification trees that do consider the ordinal value of the classes mostly assume some monotonic constraint over the explaining attributes – an assumption that does not hold in many real-life cases.

In a series of experiments based on known datasets, a comparative analysis was performed. This showed that the proposed weighted-ordinal decision tree significantly outperforms its non-ordinal counterpart, when the classification target is ordinal and there are no monotonic constraints on the explaining attributes, for both the 3-classes problems as well as the 5-classes problems. The proposed tree achieves higher precision and recall measures for boundary classes of the 3-classes problems (tagged as 'Low' and 'High') than for the central class. This observation is important since boundary classes are often associated with higher/lower risks/gains in an ordinal classification setting, such as an investment portfolio. They are suited to a decision-making process where one branch is selected by the decision maker. For the 5-classes problems the proposed tree achives higher precision and recall measures for all classes in 89% of the cases. The results of this study seem promising, however there are two important assumptions regarding the allocated weights to the different classes that need to be carefully considered. First, in order to obtain a distribution with lower WH relative to the WH of the prior distribution, it is often required that the class-mode's probability will be higher than the probability of that class in the prior distribution. Thus, initiating the algorithm with equi-probable distribution over the classes will result with more nodes in the decision tree. Second, recall that the weight of each class is normalized relative to the weights of all the other classes and does not reflect the value deviation between this class and the class-mode independently from the others. Therefore, to minimize the effect of extreme class values and obtain a more consistent weighted entropy it is preferred that the class values will be symmetrically distributed around the mode value.

Future research can address complementary calculations of the weights over the different classes, to better handle the limiting assumptions mentioned above. Future studies can also consider integrating the entropy-weighted measures into others entropy-like based methods, such as Adaboost, CARTS, Gini impurity measure, Reinforcement learning and even Entropy based Deep Network models. Furthermore, it could be interesting to examine the advantages of using an ensemble approach based on ordinal vs. non-ordinal algorithms that could leverage the performance of these classifiers. Other studies could further explore the effects of various classification parameters, such as the number of levels of the class, the value desperations and the decision-maker risk adversity on the overall classification performance.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Gonen Singer: Conceptualization, Methodology, Formal analysis, Investigation, Writing - original draft. **Roee Anuar:** Conceptualization, Methodology, Formal analysis, Investigation, Writing - original draft. **Irad Ben-Gal:** Conceptualization, Methodology, Formal analysis, Investigation, Writing - original draft.

References

Altuntas, S., & Dereli, T. (2015). A novel approach based on DEMATEL method and patent citation analysis for prioritizing a portfolio of investment projects. *Expert Systems with Applications, 42*(3), 1003–1012.

Bacher, M., & Ben-Gal, I. (2017). Ensemble-Bayesian SPC: Multi-mode process monitoring for novelty detection. *IISE Transactions, 49*(11), 1014–1030.

Ben-David, A. (1995). Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning, 19*(1), 29–43.

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

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

Breiman, L., Friedman, J., Olshen, R., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth.

Byon, E., Shrivastava, A. K., & Ding, Y. (2010). A classification procedure for highly imbalanced class sizes.. *IIE Transactions, 42*(4), 288–303.

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

Cardoso, J. S., & Sousa, R. (2011). Measuring the performance of ordinal classification. *International Journal of Pattern Recognition and Artificial Intelligence, 25*(8), 1173–1195.

Dionisio, A., Menezes, R., & Mendes, D. A. (2006). An econophysics approach to analyse uncertainty in financial markets: An application to the Portuguese stock market. *The European Physical Journal B-Condensed Matter and Complex Systems, 50*(1-2), 161–164.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems. *The Journal of Machine Learning Research, 15*, 3133–3181.

Gaudette, L., & Japkowicz, N. (2009). Evaluation methods for ordinal classification. In *Canadian Conference on Artificial Intelligence* (pp. 207–210).

Gutierrez, P. A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., & Hervas-Martinez, C. (2016). Ordinal regression methods: Survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering, 28*(1), 127–146.

Hu, Q., Che, X., Zhang, L., Zhang, D., Guo, M., & Yu, D. (2012). Rank entropy based decision trees for monotonic classification. *IEEE Transactions on Knowledge & Data Engineering, 24*(11), 2052–2064.

Hu, Q., Guo, M., Yu, D., & Liu, J. (2010). Information entropy for ordinal classification. *Science China-Information Sciences, 53*(6), 1188–1200.

Jin, R., & Deng, X. (2015). Ensemble modeling for data fusion in manufacturing process scale-up. *IIE Transactions, 47*(3), 203–214.

Mahmoud, I., & Naouib, K. (2017). Measuring systematic and specific risk: Approach mean-entropy. *Asian Journal of Empirical Research, 7*(3), 42–60.

Maimon, O., & Rokach, L. (2005). *The data mining and knowledge discovery handbook*. Heidelberg: Springer.

Marsala, C., & Petturiti, D. (2013). *Monotone classification with decision trees* (pp. 810–817). The conference of the European Society for fuzzy logic and technology.

Masetic, Z., & Subasi, A. (2016). Congestive heart failure detection using random forest classifier. *Computer methods and programs in biomedicine, 130*, 54–64.

Mouroo, R. N., Carvalho, R. S., Carvalho, R. N., & Ramos, G. N. (2017). Predicting waiting time overflow on bank teller queues. *In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 842–847.

Nabi, F. G., Sundaraj, K., Lam, C. K., & Palaniappan, R. (2019). Characterization and classification of asthmatic wheeze sounds according to severity level using spectral integrated features. *Computers in biology and medicine, 104*, 52–61.

Nawrocki, D. N., & Harding, W. H. (1986). State-value weighted entropy as a measure of investment risk. *Applied Economics, 18*(4), 411–419.

Ormos, M., & Zibriczky, D. (2014). Entropy-based financial asset pricing. *PLoS One, 9*(12), e115742.

Philippatos, G., & Wilson, C. (1972). Entropy risk and the selection of efficient portfolios. *Applied Economics, 4*, 209–220.

Potharst, R., & Bioch, J. C. (2000). Decision trees for ordinal classification. *Intelligent Data Analysis, 4*(2), 97–111.

Potharst, R., & Feelders, A. J. (2002). Classification trees for problems with monotonicity constraints. *ACM SIGKDD Explorations Newsletter, 4*(1), 1–10.

Qian, Y. H., Xu, H., Liang, J. Y., Liu, B., & Wang, J. T. (2015). Fusing monotonic decision trees. *IEEE Transactions on Knowledge and Data Engineering, 27*(10), 2717–2728.

Quinlan, R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann Publishers.

Sanit-in, Y., & Saikaew, K. R. (2019). Prediction of waiting time in one stop service. *International Journal of Machine Learning and Computing, 9*(3).

Senderovich, A., Weidlich, M., Gal, A., & Mandelbaum, A. (2015). Queue mining for delay prediction in multi-class service processes. *Information Systems, 53*, 278–295.

Shannon, C. (1948). A note on the concept of entropy. *Bell System Technical Journal, 27*, 379–423.

Shannon, C., & Weaver, W. (1949). *The Mathematical Theory of Communication*. Urbana, USA: The University of Illinois Press.

Shiju, S., & Remya, R. N. (2014). Comparative analysis of decision tree algorithms: ID3, C4.5 and Random Forest. *Computational Intelligence in Data Mining, 1*, 549–562.

Singer, G., & Golan, M. (2019). Identification of subgroups of terror attacks with shared characteristics for the purpose of preventing mass-casualty

attacks: a data-mining approach. *Crime Science, 8*(1), 14. doi:10.1186/s40163-019-0109-9.

Singer, G., Golan, M., Rabin, N., & Kleper, D. (2019). Evaluation of the effect of learning disabilities and accommodations on the prediction of the stability of academic behaviour of undergraduate engineering students using decision trees. *European Journal of Engineering Education*, 1–17.

Verbeke, W., Martens, D., & Baesens, B. (2017). RULEM: A novel heuristic rule learning approach for ordinal classification with monotonicity. *Applied Soft Computing, 60*, 858–873.

Zhao, Y., Shrivastava, A. K., & Tsui, K. L. (2016). Imbalanced classification by learning hidden data structure. *IIE Transactions, 48*(7), 614–628.

Zhu, H., Tsang, E. C. C., Wang, X. Z., & Ashfaq, R. A. R. (2017). Monotonic classification extreme learning machine. *Neurocomputing, 225*, 205–213.