2010 Ninth International Conference on Machine Learning and Applications

Classification Models with Global Constraints for Ordinal Data

Jaime S. Cardoso, Ricardo Sousa
INESC Porto, Faculdade de Engenharia, Universidade do Porto
Portugal
{*jaime.cardoso,rsousa*}*@inescporto.pt*

***Abstract*—Ordinal classification is a form of multi-class classification where there is an inherent ordering between the classes, but not a meaningful numeric difference between them. Although conventional methods, designed for nominal classes or regression problems, can be used to solve the ordinal data problem, there are benefits in developing models specific to this kind of data.**

This paper introduces a new rationale to include the information about the order in the design of a classification model. The method encompasses the inclusion of consistency constraints between adjacent decision regions. A new decision tree and a new nearest neighbour algorithms are then designed under that rationale. An experimental study with artificial and real data sets verifies the usefulness of the proposed approach.

***Keywords*-Classification; ordinal data; decision tree; k-nearest neighbour**

I. Introduction

Predictive learning has traditionally been a standard inductive learning, where different subproblem formulations have been identified. One of the most representatives is classification, consisting on the estimation of a mapping from the feature space into a finite class space. Depending on the cardinality of the finite class space we are left with binary or multiclass classification problems. Finally, the presence or absence of a "natural" order among classes will separate nominal from ordinal problems.

Although two-class and nominal classification problems have been dissected in the literature, only recently the ordinal sibling started receiving the same level of attention, both in the design of new learning formulations [1], [2] and in the development of new assessment metrics [3], [4].

In this paper we first present a novel rationale to capture and impose the order constraints in the design of a supervised classifier. The proposed formulation tries to objectify the imprecise notion of natural order. A second contribution of this paper lies on the instantiation of that underlying principle in the design of a new decision tree and a new nearest neighbour algorithms.

II. Related work

Some of the work on decision trees for ordinal data consider problems that are monotone, i.e., all attributes have ordered domains and if $\mathbf{x}, \mathbf{z}$ are data points such that $\mathbf{x} \leq \mathbf{z}$ ($x_i \leq z_i$ for each attribute i) then their labels should satisfy $\lambda(x) \leq \lambda(z)$, where $\lambda(.)$ is the labelling function: the labelling is monotone. Potharst [5], [6], [7] proposes a method that induces a binary decision tree from a monotone dataset. Methods were also proposed for non-monotone datasets (the most likely scenario in the presence of noise) but the resulting tree may be non-monotone. We will argue later the monotonicity is probably not the best way of capturing the order relationships.

Kramer et al. [8] investigate the use of a learning algorithm for regression tasks—more specifically, a regression tree learner—to solve ordinal classification problems. In this case each class needs to be mapped to a numeric value. Kramer et al. [8] compare several different methods for doing this. However, if the class attribute represents a truly ordinal quantity—which, by definition, cannot be represented as a number in a meaningful way—there is no principled way of devising an appropriate mapping and this procedure is necessarily *ad hoc*.

Frank and Hall [9] presented a simple method that enables standard classification algorithms to make use of ordering information in class attributes. By applying it in conjunction with a decision tree learner, the authors show that it outperforms the naive approach, which treats the class values as an unordered set. Compared to special-purpose algorithms for ordinal classification the method has the advantage that it can be applied without any modification to the underlying learning scheme. The rationale encompasses using $(K-1)$ standard binary classifiers to address the K-class ordinal data problem. Toward that end, the training of the i-th classifier is performed by converting the ordinal dataset with classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$ into a binary dataset, discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$. To predict the class value of an unseen instance, the $(K-1)$ outputs are combined to produce a single estimation. Any binary classifier can be used as the building block of this scheme. Observe that the $(K-1)$ classifiers are trained in an independent fashion. This independence is likely to lead to intersecting boundaries, a topic to which we will return further on in this paper.

The work on k-nearest neighbour for ordinal data seems even scarcer. Besides the well-known adaptation of labelling the test data with the median instead of the mode of the k labels, the only work the authors are aware is the modified nearest neighbour algorithm for the construction of monotone classifiers from data [10]. Again, this work continues to be limited by the assumption of monotonicity

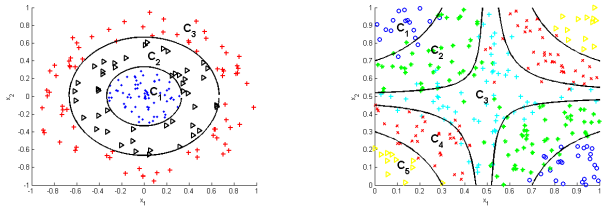
978-0-7695-4300-0/10 $26.00 © 2010 IEEE
DOI 10.1109/ICMLA.2010.18

in the input data.

We argue that current algorithms fail to incorporate appropriately the order information of the data, either because of too restrictive or too loose assumptions. The order information is a global property, i.e., it involves a relation between all data, and should therefore be the result of optimizing some global function.

III. Capturing the order constraints between classes

Assume that examples in a classification problem come from one of K ordered classes, labelled from $\mathcal{C}_1$ to $\mathcal{C}_K$, corresponding to their natural order. Unlike the monotone learning problem, where both the input attributes and the class attribute are assumed to be ordered, the setting considered in this work does not assume that the inputs are ordered. Consider the two datasets in Figure 1. The data in



(a) (b)

Figure 1. Examples of sets of ordinal data.

Figure 1(a) is uniformly distributed in the unit-circle, with the class y being assigned according to the radius of the point: $y = \left\lceil 3\sqrt{x_1^2 + x_2^2} \right\rceil$

Each point in Figure 1(b) was assigned a class y from the set $\{1, 2, 3, 4, 5\}$, according to

$$y = \min_{r \in \{1,2,3,4,5\}} \{r : b_{r-1} < 10(x_1 - 0.5)(x_2 - 0.5) + \varepsilon \ < b_r\}$$
$$(b_0, b_1, b_2, b_3, b_4, b_5) = (-\infty, -1, -0.1, 0.25, 1, +\infty) \quad (1)$$

where $\varepsilon \sim N(0; 0.125^2)$ simulates the possible existence of error in the assignment of the true class to $\mathbf{x}$.

In neither of the datasets the monotonicity constraint is verified; however, we argue that these datasets are perfectly representatives of an ordinal setting, where the order is not captured directly in the input space, but in an implicit feature space. In fact the dataset in Figure 1(b) has been used to validate algorithms for ordinal data classification [11], [1].

How to capture then the order relation in the output? Let $f(\mathbf{x})$ be a decision rule that assigns each value of $\mathbf{x}$ to one of the available classes[1]. Such a rule will divide the input space into regions $\mathcal{R}_k$ called decision regions, such that all points in $\mathcal{R}_k$ are assigned to class $\mathcal{C}_k$. The boundaries between decision regions are called decision boundaries or decision surfaces. Note that each decision region need not be contiguous but could comprise any number of disjoint regions. Intuitively, for ordinal data, in a sufficiently small neighbourhood of $\mathbf{x}$, $\mathcal{V}_\epsilon(\mathbf{x})$, the decision function should only take at most two consecutive values: $\max f(\mathbf{x}) - \min f(\mathbf{x}) \leq 1$. The motivation for this is that a small change in the input data should not lead to a 'big jump' in the output decision. Therefore, we say that a decision function is *consistent* with an ordinal data classification setting in a point $\mathbf{x}_0$ if $\exists \epsilon > 0 \ \forall \mathbf{x} \in \mathcal{V}_\epsilon(\mathbf{x}_0) \ \max f(\mathbf{x}) - \min f(\mathbf{x}) \leq 1$. A decision function is consistent in the whole input space if the above condition is verified for every point in the input space: $\forall \mathbf{x}_0 \exists \epsilon > 0 \ \forall \mathbf{x} \in \mathcal{V}_\epsilon(\mathbf{x}_0) \ \max f(\mathbf{x}) - \min f(\mathbf{x}) \leq 1$. [2]

Decision functions consistent with the ordinal setting lead to the very pleasant result that a region $\mathcal{R}_i$ where one decides for $\mathcal{C}_i$ can only be adjacent to regions $\mathcal{R}_{i+1}$ and $\mathcal{R}_{i-1}$ – see Figure III.

(a) Regions consistent with the ordinal setting.

(b) Regions inconsistent with the ordinal setting.

Figure 2. Consequence of the consistency constraint in the arrangement of the decision regions.

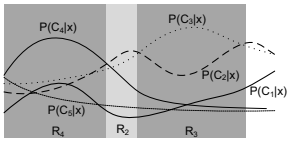
The rationale here introduced is a generalization of the formulation of parallel boundaries adopted in linear SVMs for ordinal data [12] and the non-intersecting boundaries approach adopted in [1]. We also notice that the approach by Frank and Hall [9] may lead to inconsistent solution under the adopted formulation since the design of independent classifiers will likely result in intersecting boundaries.

It is also interesting to establish a parallel with the probabilistic framework introduced previously by Pinto da Costa et al. [13]. The unimodal model assumes that for any given point $\mathbf{x}$ the posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$ follow a unimodal distribution. Given a point $\mathbf{x}$, if the highest a posteriori probability is, for instance, $p(\mathcal{C}_k|\mathbf{x})$, then we

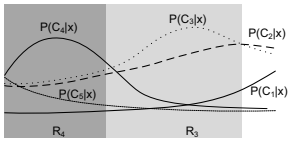
[1] A remark should be made. Since we are dealing with ordered classes, we shall consider that the output of the decision function is one of the K labels $\{\mathcal{C}_1, \cdots, \mathcal{C}_K\}$ or one number in $\{1, \cdots, K\}$ resulting from the bijective map $g : \{\mathcal{C}_i\}_{i=1}^K \to \{1, \cdots, K\}$ which assigns the number k to the class $\mathcal{C}_k$, i.e., $g(\mathcal{C}_k) = k$. The context should make it clear which of the two output formats is being considered.

[2] This definition of consistency precludes decision functions such as $f(x) = 1, x < 0; f(x) = 2, x = 0; f(x) = 3, x > 0$, where the region corresponding to class 2 is a measure-zero set.

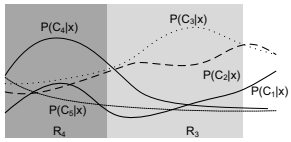
should have, given that there is an order relation between the classes, $p(\mathcal{C}_1|\mathbf{x}) < \cdots < p(\mathcal{C}_{k-1}|\mathbf{x}) < p(\mathcal{C}_k|\mathbf{x}) > p(\mathcal{C}_{k+1}|\mathbf{x}) > \cdots > p(\mathcal{C}_K|\mathbf{x})$: $\mathcal{C}_{k-1}$ and $\mathcal{C}_{k+1}$ are closer to $\mathcal{C}_k$ and therefore the second highest a posteriori probability should be attained in one of these classes, see Figure 3(b). Had one used a classifier which does not take into account the order relation between the classes, the second highest a posteriori probability can be, for instance, $p(\mathcal{C}_{k-2}|\mathbf{x})$, see Figure 3(a).



(a) Illustrative posteriori class distribution for a conventional nominal data problem.



(b) Illustrative posteriori class distribution for the unimodal model for ordinal data.



(c) Illustrative posteriori class distribution sufficient to assure the consistency property for ordinal data.

Figure 3. Illustrative posteriori class distributions for different models.

While the unimodal model imposes an order relationship between any two consecutive class probabilities, such a strict condition is not required to observe the consistency property we introduce in this work. In fact, the consistency property will be observed if the following conditions, in-between the conventional formulation for nominal data and the unimodal model, are true:

$$\begin{aligned} p(\mathcal{C}_k|\mathbf{x}) > p(\mathcal{C}_{k-1}|\mathbf{x}) > p(\mathcal{C}_i|\mathbf{x}), \quad \forall\ 1 < i < k-1 \\ p(\mathcal{C}_k|\mathbf{x}) > p(\mathcal{C}_{k+1}|\mathbf{x}) > p(\mathcal{C}_i|\mathbf{x}), \quad \forall\ k+1 < i < K \end{aligned} \tag{2}$$

Intuitively, one just needs to impose that the second higher probability is the 'right' one. This is sufficient (although not necessary) to assure that, at the decision boundaries the decision rule will change for an adjacent class.

IV. Imposing the ordinal constraints in a decision function

Consistency is a global property, i.e., it involves a relation between different decision regions of the space. A key challenge is how to use this information during the design process of a learning algorithm. In this section we consider that a decision function has already been obtained by, possibly, standard methods and use the consistency property to relabel the decision regions.

It is convenient at this point to define some notation to describe the assignment of labels to different decision regions. Let $\mathcal{R}_n$, $n = 1, \cdots, N$, represent the contiguous decision regions created by some model[3]. For each region $\mathcal{R}_n$ we introduce a corresponding set of binary indicator variables $x_{n,k} \in \{0, 1\}$, where $k = 1, \cdots, K-1$ describing which of the K ordinal labels is assigned to region $\mathcal{R}_n$, so that if data points in $\mathcal{R}_n$ are assigned the label k then $x_{n,j} = 1$ for $j < k$, and $x_{n,j} = 0$ otherwise. So, for instance if we have a setting with 5 classes, $K = 5$, and to a particular region happens to be assigned the label 3, then $\mathbf{x}$ will be represented by $\mathbf{x} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^t$. Note that this is different from the often used 1-of-K coding scheme and we find it more convenient for the introduction of the constraints in what follows.

In ordinal data settings, the loss associated with a region $\mathcal{R}_n$ when deciding for class $\mathcal{C}_k$ is usually captured with the absolute error, the sum over all points lying in $\mathcal{R}_n$ of the absolute difference between the true class of the point and the predicted class for the region:

$$c_{n,k} = \sum_{i=1}^{K} |i - k| p_{n,i},$$

where $p_{n,i}$, $n = 1, \cdots, N$, $i = 1, \cdots, K$ represent the number of observations (from the data used in creating the region by some learning algorithm) from class k satisfying the conditions for region $\mathcal{R}_n$, (that is, lying inside $\mathcal{R}_n$). Nevertheless, the following model is generic for any costs $c_{n,k}$.

The optimal labelling of the regions can then be found by minimizing the following objective function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} c_{n,k}(x_{n,k-1} - x_{n,k}), \tag{3}$$

where the *constants* $x_{n,0} = 1$ and $x_{n,K} = 0$ have been introduced for notational convenience, with the constraints

$$x_{n,k+1} - x_{n,k} \leq 0, k = 1, \cdots, K-2, \quad n = 1, \cdots, N \tag{4}$$

and

$$x_{n,k} \in \{0, 1\}, k = 1, \cdots, K-1, \quad n = 1, \cdots, N \tag{5}$$

It is easily seen that Eq. (3) can be rewritten as

$$J = \sum_{n=1}^{N} \left\{ c_{n,1} + \sum_{k=1}^{K-1} x_{n,k}(c_{n,k+1} - c_{n,k}) \right\}, \tag{6}$$

Without any constraints relating the labels of the regions, the optimization of the loss J over the whole space leads to the standard solution of predicting the median of the values in each region.

[3]Note the change of notation: so far we have used $\mathcal{R}_k$ to represent the decision region, contiguous or not, corresponding to class $\mathcal{C}_k$. From now on $\mathcal{R}_n$ just represents a continuous region of the space with all points inside that region being assigned the same class. Therefore, different regions $\mathcal{R}_n$ and $\mathcal{R}_m$ may be assigned the same class and the number of regions is likely greater than the number of classes.

Now, we want to impose that adjacent regions have labels that differ at most by one. Therefore we are led to the optimization of the loss of the decision function constrained by the consistency of it. Consistency imposes that, for any pair of adjacent regions $\mathcal{R}_n$ and $\mathcal{R}_{n'}$, the following inequality must be verified:

$$\left| (1 + \sum_{k=1}^{K-1} x_{n,k}) - (1 + \sum_{k=1}^{K-1} x_{n',k}) \right| \leq 1 \tag{7}$$

Inequality (7) can be written as

$$\begin{aligned} \sum_{k=1}^{K-1} x_{n,k} - \sum_{k=1}^{K-1} x_{n',k} \leq 1 \\ \sum_{k=1}^{K-1} x_{n',k} - \sum_{k=1}^{K-1} x_{n,k} \leq 1 \end{aligned} \tag{8}$$

The optimization of (6), subject to constraints (4), (5) and (8) constitutes a linear binary integer programming problem.

Although the resulting constraint matrix is not totally unimodular (which would allow the relaxation of the linear binary integer programming problem to a much easier linear programming problem), we found experimentally that the actual shape and sparsity of the constraint matrix of typical problems favour the efficiency of the algorithm. Nevertheless, further research on the computational complexity of the method is required.

A. Algorithms for solving the 0-1 linear model

In this section we focus on two algorithms for solving the 0-1 linear model. Although for small problems the 0-1 formulation can be used directly, this approach becomes prohibitive with the increase of the dimension of the data, the increase of the size of training set or with the increase of the number of classes.

1) Iterative algorithm: The observation that decision regions for class $\mathcal{C}_k$ are more likely to be adjacent to regions labelled for $\mathcal{C}_j$ with $|j-k|$ small, suggests a block coordinate optimization procedure, where the consistency constraints are imposed iteratively to a different subset of regions.

Initializing the region labels to the conventional value obtained from the median label of the points assigned to the region, we propose to iteratively select a subset of regions with labels in the interval $\mathcal{C}_j, \cdots, \mathcal{C}_{j+W-1}$ and re-label those regions with the output of the optimization problem restricted to those regions. The simplest solution is to simply iterate j from 1 to $K-W+1$. Note that if we select $W = K$ we would be solving the complete original problem; if we select $W = 2$ no constraint will be imposed and one stays in the solution without consistency constraints.

Note that the global consistency of the solution obtained at the end of the iterative process is not assured.

2) Approximation algorithm based on LP relaxation: A relaxation procedure starts by choosing and solving a relaxation problem for obtaining an approximated solution; then, it uses a rounding procedure to extract a feasible solution to the original 0-1 problem from the approximate solution. The relaxation step has an important role in the whole algorithm. For example, if the approximation solution is in fact feasible for the original problem, then it is exactly an optimal solution. On the other hand, when the approximation solution is not feasible regarding the original problem, we have to use a rounding procedure to extract a feasible solution.

The relaxed model for our 0-1 problem is obtained by replacing the constraint (5) by

$$x_{n,k} \in [0,1], k = 1, \cdots, K-1, \quad n = 1, \cdots, N \tag{9}$$

Solving now (6), subject to constraints (4), (9) and (8) finds the solution to our relaxed problem.

Noting now that (4), together with the monotonicity of the round function, assures that the rounded solution is a valid coding for the class — although not necessarily a feasible solution since the constraints (8) may not be observed — , that terminates the relaxation method. Again, the global consistency of the solution obtained at the end of the iterative process is not assured.

V. An ordinal decision tree

The root of the majority of the work on decision trees is in Breiman's work [14] and Quinlan's ID3 algorithm [15] from statistical and machine learning perspectives. Decision trees are hierarchical decision systems in which conditions are sequentially tested until a class is accepted. To this end, the feature space is split into unique regions, corresponding to the classes, in a sequential manner. Upon the arrival of a feature vector, the searching of the region to which the feature vector will be assigned is achieved via a sequence of decisions along a path of nodes of an appropriately constructed tree. The most popular schemes among decision trees are those that split the space into hyperrectangles with sides parallel to the axes. The sequence of decisions is applied to individual features, and the questions to be answered are of the form "is feature $x_k \leq \alpha$?" where α is a threshold value. Such trees are known as ordinary binary classification trees (OBCTs).

An algorithm for the induction of a decision tree from a training dataset contains the following ingredients:

- a splitting rule: at each node, the set of candidate questions to be asked has to be decided. Each question corresponds to a specific binary split into two descendant nodes. A splitting criterion must be adopted according to which the best split from the set of candidate ones is chosen.
- a stopping rule: A stop-splitting rule is required that controls the growth of the tree and a node is declared

as a terminal (leaf). The most commonly used approach is to grow the tree up to a large size first and then prune nodes according to a pruning criterion. A number of pruning criteria have been suggested. A commonly approach is to combine an estimate of the error probability with a complexity measuring term (e.g. number of terminal nodes) [16].

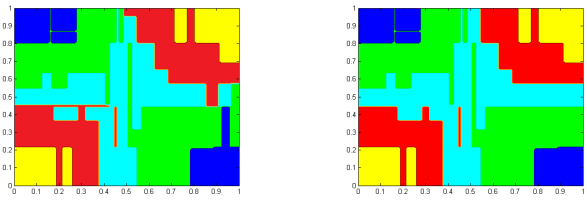
- a labelling rule: a rule is required that assigns each leaf to a specific class.

A. Imposing the ordinal constraints in a decision tree: the oTree model

If the consistency is measured for each possible split during tree construction, the order in which nodes are expanded becomes important. For example, a depth-first search strategy will generally lead to a different tree than a breadth-first search. Also, and perhaps more importantly, a non-consistent tree may become consistent after additional splits.

In view of these difficulties, in this work we consider imposing consistency only during the labelling assignment step. Future work will address other mechanisms. Consider an already constructed tree, using any standard technique such as C4.5 [17], perhaps already pruned according to a pre-specified strategy.

We can now apply the rationale developed in the previous section to the regions corresponding to each leaf of the tree. In this scenario, each region is a hyperrectangle. In Figure 4 is depicted the decision regions obtained by growing a tree without pruning from 300 random observations generated according to Eq. (1). In Figure 4(b) is visible the benefits of imposing the consistency constraints by relabeling the leaves. It is also interesting to interpret the consistency constraints as a regularization factor in the tree building process.



(a) Without consistency constraints. (b) With consistency constraints.

Figure 4. Decision regions for a fully-grown tree with 300 random observations generated according to Eq. (1).

VI. An ordinal k Nearest-Neighbour: the okNN model

The k-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms. This algorithm belongs to a set of techniques called Instance Based Learning. It starts by extending the local region around a data point until the k^{th} nearest neighbour is found. For nominal data, an object is classified by a majority vote scheme, with the object being assigned to the class most common amongst its k-nearest neighbours; for ordinal data, the median is usually preferred.

In the simplest case, consider $k = 1$ and a given set of points S. Each training point $\mathbf{x}_i$ defines a Voronoi cell R_i, a convex polytope, consisting of all points closer to $\mathbf{x}_i$ than to any other training point $\mathbf{x}_j$. The label assigned to a given Voronoi cell R_i is the label of the corresponding training point $\mathbf{x}_i$.

The consistency constraints for ordinal data introduced before are also easily integrated in the 1-NN classifier. Now the regions involved in the optimization process are the Voronoi cells; the cost $c_{n,k}$ is simply $c_{n,k} = |k - i|$, where i is the class of the training point in the cell. The adjacency can be tested by testing the adjacency of the corresponding polytopes.

The extension to the k-NN can be accomplished in two ways. One option is to apply the consistency constraints directly on the generalized Voronoi cells corresponding to the k-NN as a post-processing, identically to what was just proposed for the 1-NN. Another option is to use the above procedure on 1-NN as a pre-processing before applying a standard k-NN. It is possible to show that, under some conditions, the resulting decision function is consistent.

Consider the neighbourhood $V_k(\mathbf{x})$ containing the k nearest training points of the (test) point $\mathbf{x}$. Let m be the minimum and M the maximum of those k labels. Under the assumption that the training points have been relabelled by imposing the consistency constraints in the 1-NN classifier, the set of the k labels contains every label between m and M. Consider the Voronoi cells from 1-NN that intersect $V_k(\mathbf{x})$ and a graph with a vertex in each of the k training points and an edge for each pair of adjacent training points (for which the cells are adjacent). Then there is a path between any pair of vertices, and in particular between a point labelled with m and a point labelled with M. Since the Voronoi cells are consistent, the path must go throw each possible label between m and M. Now, adjacent regions in the k-NN differing at a single of the k points will then also differ at most by one in the median of the k points. When adjacent regions differ at more than 1 of the k point due to, for instance, coincident training points, the consistency is not assured.

VII. Experimental study

We started by conducting an empirical comparison in an artificial dataset between a standard classification tree (cTree), a standard kNN and the oTree and okNN models proposed in this work. The comparison study is based on the Mean Absolute Error (MAE), which is the most commonly used for ordinal data. The experimental study was conducted

in Matlab R2009b. The conventional tree model was based on the `classregtree` class, with the labelling rule adapted to use the median of the values instead of the mode. The kNN used the `knnclassify` function.

We began by generating 1000 examples from the dataset presented in Section III, given by Eq. (1), and randomly split 50 times the generated dataset into training and test sets. Each model parameterization, namely the pruning level of the tree and the size k of the neighbourhood of kNN was selected by 5-fold cross-validation on the training set. Results were averaged over the 50 setups in order to get more robust estimates. This was repeated taking $\ell \in \{100, 300, 500\}$ for size of the training set and $1000 - \ell$ for the test set size. The small size of the dataset allowed us to use directly the 0-1 exact formulation for the relabeling procedure. The test results for are shown in Table I. It can be seen that there

	Training sets size		
Model	$\ell = 100$	$\ell = 300$	$\ell = 500$
cTree	0.47 (0.11)	0.30 (0.05)	0.22 (0.03)
oTree	0.40 (0.10)	0.27 (0.04)	0.22 (0.02)
kNN	0.29 (0.03)	0.24 (0.02)	0.22 (0.02)
okNN	0.28 (0.02)	0.23 (0.02)	0.21 (0.01)

Table I
MEAN (STANDARD DEVIATION) OF MER OVER 50 SETUPS OF THE SYNTHETIC DATASET.

are no significant differences between the conventional and the proposed models, with only a slightly advantage for the latter. Nevertheless, the proposed models also show higher stability (lower variance) and produce smaller and consistent models.

We continue the experimental study by applying the algorithms under evaluation to the classification of real data, mostly available on the Weka datasets website.[4] and UCI machine learning repository. The SWD dataset contains real-world assessments of qualified social workers regarding the risk facing children if they stayed with their families at home and is composed by 10 features and 4 classes. The LEV dataset contains examples of anonymous lecturer evaluations, taken at the end of MBA courses and is composed by 4 features and 5 classes. Both datasets contain 1000 examples. The third dataset, BCCT, encompasses 960 observations taken from our previous work [18] and expresses the aesthetic evaluation of Breast Cancer Conservative Treatment. For each patient submitted to BCCT, 30 measurements were recorded, capturing visible skin alterations or changes in breast volume or shape. Only the 5 features selected in [18] were used in the experimental work. The aesthetic outcome of the treatment for each and every patient was classified in one of the four categories: Excellent, Good, Fair and Poor. Finally, the Diabetes dataset represents a regression prediction problem converted to an ordinal quantity using equal-frequency binning, dividing the range of observed values into 5 intervals so that the number of instances in each interval is approximately constant. The dataset includes 43 observations with 2 features. The test results are shown in Table II, for the MER criterion.

[4]for more information, please see: http://www.cs.waikato.ac.nz/~ml/weka/index_datasets.html.

	Datasets			
Model	SWD	LEV	BCCT	Diabetes
cTree	0.48 (0.03)	0.45 (0.02)	0.45 (0.04)	0.71 (0.05)
oTree	0.47 (0.03)	0.45 (0.02)	0.42 (0.05)	0.68 (0.04)
kNN	0.57 (0.03)	0.58 (0.05)	0.53 (0.04)	0.78 (0.06)
okNN	0.57 (0.04)	0.56 (0.04)	0.54 (0.02)	0.72 (0.04)

Table II
MEAN (STANDARD DEVIATION) OF MER OVER 50 SETUPS OF THE DATASETS.

Again, the same relative behaviour is observed in these real datasets. It is also visible that the decision tree usually attains better results than the k-nearest neighbour. Even if the proposed framework seems to help improve the performance of a model, that did not always happen. We conjecture that the use of the consistency property only as a post-processing operation may lead to 'over-regularized' or over-smoothed decision functions, effectively hurting or attenuating the positive impact on the generalization performance of the model.

Although this has been a limited experimental study, it provides a first validation of the proposed method. The proposed method is likely to produce simpler, consistent and easier to interpret models. Further experiments, including large datasets, are required and will be conducted in a future research.

VIII. CONCLUSIONS

We have provided a new rationale for the incorporation of the order information in the design of classification models intended for ordinal data. The fundamental idea is that adjacent decision region should have equal or consecutive labels. The rationale was then used as a post-processing mechanism of a standard decision tree and as a pre- or post-processing step for the k-NN. We have conducted several experiments where our method was tested against standard models from where our method was derived. The results show some advantages of the proposed method.

These initial investigations support further work in this direction. In future extensions of this work, in addition to a stronger experimental validation, we intend to quantify the regularization effect of the ordinal constraints and the generalization bounds of the method. We will also analyse the computational efficiency of the binary optimization procedure involved in the relabeling of the leaves. Extensions of the work may encompass the adaptation of the pruning or splitting strategies of tree models. Dyadic trees [19] may provide an adequate environment to research some of the

previous topics. In fact, although the proposed consistency underlying principle has been applied as a pre- and post-processing of the result of a standard method, nothing prevents its application during the design of the decision model. The connection established with the unimodal model may provide some suggestions in that direction.

ACKNOWLEDGMENT

This research was partially supported by the Carnegie Mellon | Portugal Program and conducted when J.S.C. was a Visiting Professor at Carnegie Mellon University under the Faculty Exchange Program of the Carnegie Mellon | Portugal Program. The authors would like to thank Eric P. Xing for the support in this work. This research was also partially supported by Fundação para a Ciência e a Tecnologia (FCT) - Portugal through project PTDC/EIA/64914/2006.

REFERENCES

[1] J. S. Cardoso and J. F. P. da Costa, "Learning to classify ordinal data: the data replication method," *Journal of Machine Learning Research*, vol. 8, pp. 1393–1429, 2007.

[2] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *Journal of Machine Learning Research*, vol. 6, pp. 1019–1041, 2005.

[3] L. Gaudette and N. Japkowicz, "Evaluation methods for ordinal classification," in *Proceedings of the 2nd Canadian Conference on Artificial Intelligence*, ser. Lecture Notes in Computer Science, Y. Gao and N. Japkowicz, Eds. Springer, 2009, pp. 207–210.

[4] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal regression," in *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications*, 2009, pp. 283–287.

[5] R. Potharst and J. C. Bioch, "A decision tree algorithm for ordinal classification," in *Advances in Intelligent Data Analysis*, 1999, pp. 187–198.

[6] ——, "Decision trees for ordinal classification," *Intelligent Data Analysis*, vol. 4, no. 2, pp. 97–111, 2000.

[7] R. Potharst and A. J. Feelders, "Classification trees for problems with monotonicity constraints," *SIGKDD Explorations Newsletter*, vol. 4, no. 1, pp. 1–10, 2002.

[8] S. Kramer, G. Widmer, B. Pfahringer, and M. D. Groeve, "Prediction of ordinal classes using regression trees," *Fundamenta Informaticae*, pp. 1–13, 2001.

[9] E. Frank and M. Hall, "A simple approach to ordinal classification," in *ECML '01: Proceedings of the 12th European Conference on Machine Learning*. London, UK: Springer-Verlag, 2001, pp. 145–156.

[10] W. Duivesteijn and A. Feelders, "Nearest neighbour classification with monotonicity constraints," in *ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 301–316.

[11] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," in *Ninth International Conference on Artificial Neural Networks ICANN*, vol. 1, 1999, pp. 97–102.

[12] A. Shashua and A. Levin, "Ranking with large margin principle: Two approaches," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 937–944.

[13] J. F. P. da Costa, H. Alonso, and J. S. Cardoso, "The unimodal model for the classification of ordinal data," *Neural Networks*, vol. 21, pp. 78–91, 2008.

[14] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Belmont, California: Wadsworth, 1984.

[15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[16] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge Press, 1986.

[17] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Belmont, California: Morgan Kaufmann Publishers, 1993.

[18] J. S. Cardoso and M. J. Cardoso, "Towards an intelligent medical system for the aesthetic evaluation of breast cancer conservative treatment," *Artificial Intelligence in Medicine*, vol. 40, pp. 115–126, 2007.

[19] C. Scott and R. D. Nowak, "Minimax-optimal classification with dyadic decision trees," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1335–1353, 2006.