Ensemble of Decision Trees with Global Constraints for Ordinal Classification

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

***Abstract*—While ordinal classification problems are common in many situations, induction of ordinal decision trees has not evolved significantly. Conventional trees for regression settings or nominal classification are commonly induced for ordinal classification problems. On the other hand a decision tree consistent with the ordinal setting is often desirable to aid decision making in such situations as credit rating.**

In this work we extend a recently proposed strategy based on constraints defined globally over the feature space. We propose a bootstrap technique to improve the accuracy of the baseline solution. Experiments in synthetic and real data show the benefits of our proposal.

***Keywords*-Decision Trees, Supervised Learning, Classification, Ordinal Data, Ensemble Learning**

I. Introduction

Machine learning is playing a central role in deployment of the so-called intelligent systems, where inductive learning techniques are usually used to induce a general rule from a set of observed instances. Among the wide family set of inductive learning schemes, classification is of fundamental importance.

With classification one is interested on finding a mapping from a point (or observation) in $\mathbb{R}^d$ to a value from a finite set. The first is usually called feature space, whereas the latter is called output space. Depending on the problem, this output space can be composed by a set of only two or $K > 2$ elements, the binary or multiclass problem, respectively. The multiclass problem can be further subdivided into the nominal and the ordinal problem. For instance, problems like credit scoring where the system evaluates the capability of one default his debts, or gene analysis through the analysis of hyperactivity on certain proteins, are some examples of ordinal problems where data is structured by a "natural" order. A concrete example would be the grading of a customer credit profile in the scale Excellent $\succ$ Good $\succ$ Fair $\succ$ Poor.

Ordinal classification has evolved considerably in recent years, receiving significant attention from the scientific community. One of the reasons for this trend is probably due to its wide range of applications on the real world problems, which identified this topic as a very rich niche in the machine learning field. Conventional methods for nominal classes or for regression problems can and have been used to solve the classification of ordinal data. However, the use of specific methods for ordered classes has the potential of resulting in simpler classifiers with better generalisation capability, and the potential of facilitating the identification of the factors most influencing class discrimination.

Imposing ordinality during the model construction of interpretable learning schemes like Decision Trees (DTs) is not straightforward. DTs are well established in the scientific community, being naturally endowed with the interpretable feature. Interpretable models are, in some environments, an asset in the way that they allow the decision maker to understand the suggested decision instead of just having the decision. Such environments are usually referred, but not restrict to, medical systems where a computer aided diagnosis systems are developed to analyse patient data in order to aid the expert. In [1], [2] it is proposed an learning system for ordinal data based on DTs where it is assumed that features and classes are linearly ordered. More precisely, the authors assume that if $\mathbf{x}_1$ and $\mathbf{x}_2$ are any two observations such that $\mathbf{x}_1 \leq \mathbf{x}_2$ (for every component of $\mathbf{x}_1$ and $\mathbf{x}_2$) then it will also be true that $f(\mathbf{x}_1) \leq f(\mathbf{x}_2)$, being f the model. Frank and Hall [3] adopted a different approach, proposing the use of $(K-1)$ standard binary classifiers to address the K-class ordinal data problem. [4], [5] recovered this problem on DTs systems using either regression or evaluation metrics to attain better classifiers in the ordinal setting. In [6] it was proposed to impose ordinality after the training taking place through regions relabelling. The fundamental idea is that adjacent decision regions should have equal or consecutive labels. This strategy was instantiated in two classical learning schemes: DTs and nearest neighbour (NN). Here we recover the work proposed in [6] in order to diminish the over-regularisation issue identified by the authors. Through the usage of ensemble learning techniques, we can fuse the set of resultant trees into a single one. By applying a new formulation for the global constraints in order to impose the order, we can avoid over-regularised output decision regions.

In Section II we recover the global constraints concept. Afterwards, we present our approach delving a new formulation for the strategy in consideration. In Section III an evaluation is performed over synthetic and real datasets, concluding with some remarks in Section IV.

II. Imposing Ordinality on Decision Trees

Different studies have proposed different adaptations to learning schemes to cope with the ordinality setting, ranging from support vector machines, to neural networks or Gaussian Processes. Hierarchical models, like DTs, pose a difficult challenge: the fact that they are designed as a sequence of local decisions raises difficulties when trying to incorporate the information about the order in the learning process. Some initial efforts include the already mentioned work by Potharst [1], [2] through the construction of monotone decision trees or the simple approach by Frank and Hall [3]. In [6] we paved the way towards a more generic setting for these kind of problems, arguing that the order information is a global property, i.e., it involves a relation between all data, and should therefore be the result of optimising some global function. In Section II-A we will recall some key concepts introduced on our previous work and motivate the extension proposed here.

A. Consistency

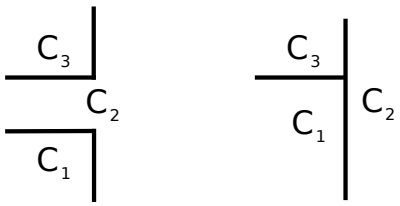
When considering an ordinal problem, for instance, assessing a customer credit score on a scale Excellent $\succ$ Good $\succ$ Fair $\succ$ Poor, there is some knowledge about the possible labels and some dependencies between them that would be interesting to model. How to capture then the order relation in the output? In [6] we tackled this problem with the introduction of the notion of consistency with the ordinal setting, which we describe here for completeness.

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]

[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.

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 1.



(a) Regions consistent with the ordinal setting. (b) Regions inconsistent with the ordinal setting.

Figure 1: Consequence of the consistency constraint in the arrangement of the decision regions.

1) 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:

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

[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.

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 minimising the following objective function

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

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

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

and

$$x_{n,k} \in \{0, 1\}, k = 1, \cdots, K-1, \quad n = 1, \cdots, N \tag{3}$$

It is easily seen that Eq. (1) can be rewritten as

$$J = \sum_{n=1}^{N} \left\{ c_{n,1} + \sum_{k=1}^{K-1} x_{n,k}(c_{n,k+1} - c_{n,k}) \right\}, \tag{4}$$

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

Now, we want to impose that adjacent regions have labels that differ at most by one. Therefore we are led to the optimisation 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:

$$\left| (1 + \sum_{k=1}^{K-1} x_{n,k}) - (1 + \sum_{k=1}^{K-1} x_{n',k}) \right| \leq 1 \tag{5}$$

Inequality (5) can be written as

$$\begin{aligned} \sum_{k=1}^{K-1} x_{n,k} - \sum_{k=1}^{K-1} x_{n',k} \leq 1 \\ \sum_{k=1}^{K-1} x_{n',k} - \sum_{k=1}^{K-1} x_{n,k} \leq 1 \end{aligned} \tag{6}$$

The optimisation of (4), subject to constraints (2), (3) and (6) constitutes a linear binary integer programming problem.

2) Avoiding Over-Regularised Decision Spaces: Even if this baseline framework has the potential to improve the performance of a model, that did not always happen in the experiments reports in [6]. We conjecture that the use of the consistency property only as a post-processing operation may lead to 'over-regularised' or over-smoothed decision functions, effectively hurting or attenuating the positive impact on the generalisation performance of the model. This over-regularisation could be especially true with small datasets, precisely when it is more needed.

One way to try overcoming this problem is to force an over-partition of the space prior to the relabelling for global consistency. One would expect that the global optimisation would then compensate this initial over-refinement. Resampling techniques [7], noise induction [8], or other similar approaches could be used to induce this over-partition of the space. In here we explore the resampling approach on the context of ensemble learning.

Although the bootstrap technique is a general tool for assessing statistical accuracy, it can also be used to improve the accuracy of a prediction scheme. The basic idea is to randomly draw datasets with replacement from the training data, each sample the same size as the original training set. This is done B times ($B = 100$ say), producing B bootstrap datasets. Then we fit a DT to each of the bootstrap datasets. Typically bootstrap aggregation or bagging would then select the class with the most "votes" from the B DTs. In here we will consider the option of working directly with the partition of the space corresponding to each DT—see Figure 2.

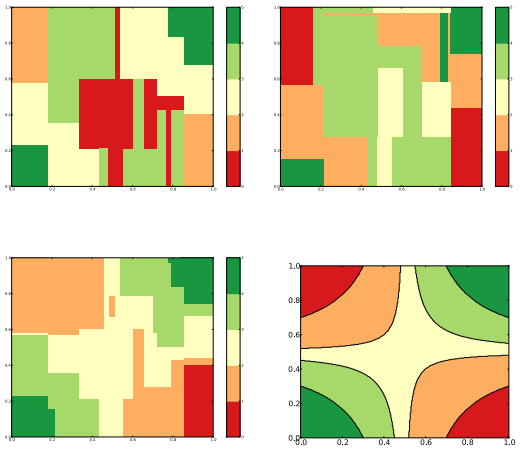


Figure 2: Example of individual models and their aggregation under an ensemble algorithm. Top figures: two distinct models; bottom left figure: aggregated regions of the two models; bottom right figure: optimal decision boundaries.

Instead of bagging directly the output of the B DTs we propose to group first the B DTs in groups of M DTs and to compute the fusion (intersection) of the M corresponding space partitions, see Figure 3. Each fused partition will then be relabelled according to the consistency optimisation procedure described earlier. Finally, we bag the relabelled models. Since we are dealing with ordinal data, we use the median of the B/M votes as the final decision. A natural question to ask is if the model induced by the bagging procedure is still consistent according to

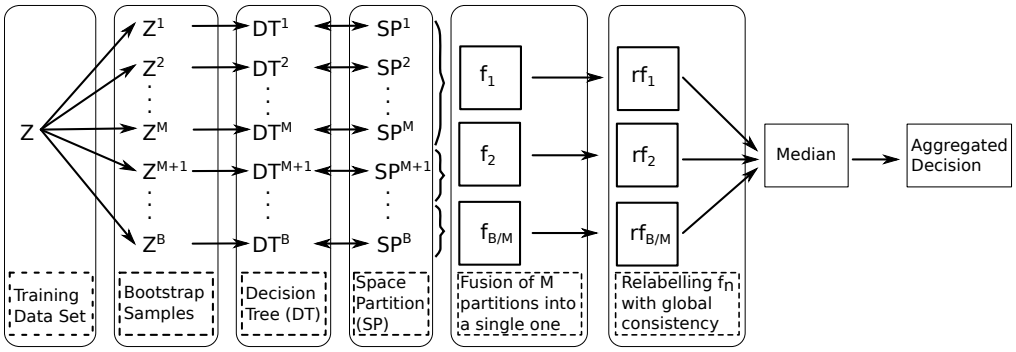


Figure 3: Schematic of the proposed aggregation process.

our previous definition. That this is indeed true is easily confirmed.

Theorem 1: Aggregation of consistent decisions produces a consistent decision when using the 'median voting' as the fusion rule.

Proof: Consider $\mathbf{x}$ and the $L = B/M$ predictions $y_1, \cdots, y_L$ at $\mathbf{x}$ by the L models, which are by construction consistent. Consider $\mathbf{x}+\delta$ in a small enough neighbourhood of $\mathbf{x}$ so that the L predictions $z_1, \cdots, z_L$ at $\mathbf{x}+\delta$ obey the consistency constraint, namely $z_i \in \{y_i - 1, y_i, y_i + 1\}$. The consistency of the 'median voting' scheme results from the simple observation that since $y_i - 1 \leq z_i \leq y_i + 1$ then $\text{median}(y_1, \cdots, y_L) - 1 = \text{median}(y_1 - 1, \cdots, y_L - 1) \leq \text{median}(z_1, \cdots, z_L) \leq \text{median}(y_1 + 1, \cdots, y_L + 1) = \text{median}(y_1, \cdots, y_L) + 1$. ■

Global consistency with empty regions: The fusion mechanism is likely to produce empty regions, i.e., regions without instances from the training set. A direct consequence is that the optimisation procedure provided early becomes ill-defined, in the sense that there are multiple optimal labellings. In fact, any relabelling of the empty regions that is still consistent does not change the value of the objective function, see Figure 4. We set additional constraints on

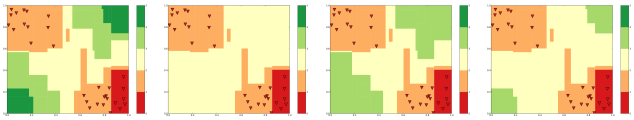


Figure 4: Different labelling with the same value for the optimisation function (objective function in Equation (4) s.t. (2), (3) and (6)).

the labels of the empty regions so that the optimisation problem becomes again well defined. Intuitively, we argue that adjacent empty regions would share the same label. Instead of forcing hard constraints, we suggest to penalise in the objective function any deviation of this goal. The constraints given in Equation (6) are re-written for pairs of regions involving empty regions as in Equation (7):

$$\begin{aligned} \sum_{k=1}^{K-1} x_{n,k} - \sum_{k=1}^{K-1} x_{n',k} \leq \delta_{(n,n')} \\ \sum_{k=1}^{K-1} x_{n',k} - \sum_{k=1}^{K-1} x_{n,k} \leq \delta_{(n,n')} \end{aligned} \qquad \forall (n,n') \in \Delta \quad (7)$$

$$\delta_{(n,n')} \in \{0,1\} \qquad \forall (n,n') \in \Delta \quad (8)$$

where Δ contains all empty adjacent regions. The objective function is also updated with a regularisation factor as represented in Equation (9):

$$J = \sum_{n=1}^{N} \left\{ c_{n,1} + \sum_{k=1}^{K-1} x_{n,k}(c_{n,k+1} - c_{n,k}) \right\} + C \sum_{\forall (n,n') \in \Delta} \delta_{(n,n')}, \quad (9)$$

where $C > 0$ controls the tradeoff between the smoothness over the labels of the empty regions, which we want to impose and the need to satisfy the consistency property. Since the new term in the objective function has the single purpose of, among the solutions satisfying the consistency property, favour the solutions with 'almost' constant labels in the empty regions, C should be 'sufficiently' small so that inconsistent solutions (but very smooth over the empty regions) are not preferred. However, in this formulation, pairs of adjacent regions where both are empty and pairs which have exactly one empty region are treated equally in terms of the relabelling cost. Take for instance the possible

Case 1:	$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_3$
Case 2:	$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_3$
Case 3:	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_3$

Table I: Different possible labellings.

labellings in Table I. Assume that the decision regions on the first and last columns are populated with some instances of the training set whereas the remaining decision regions are empty. The training observations in the first and last

columns are such that the optimal decision is those regions is $\mathcal{C}_1$ and $\mathcal{C}_3$, respectively.

All three labelling are equivalent by the baseline optimisation criterion [6]. However, the last two are preferred over the first one by the re-formulation in Equations (7),(8),(9), since both minimise the number of label transitions.

Intuitively, empty regions adjoin with non-empty regions should share the label of the non-empty region. The rationale is similar to the margin maximisation of other learning schemes, putting the transition between labels further away from the data points. Therefore, pairs of empty regions should have a lower penalty than pairs which have exactly one empty region.

Letting Δ_1 be the set containing only pairs of empty regions and Δ_2 the set of pairs which have exactly one empty region[4], we penalise differently the deviation of the aforementioned objective:

$$J = \sum_{n=1}^{N} \left\{ c_{n,1} + \sum_{k=1}^{K-1} x_{n,k}(c_{n,k+1} - c_{n,k}) \right\} + C_1 \sum_{\forall (n,n') \in \Delta_1} \delta_{(n,n')} + C_2 \sum_{\forall (n,n') \in \Delta_2} \delta_{(n,n')}, \quad (10)$$

with $C_2 > C_1 > 0$. We defined C_1 with value of $1/(N(K-1))$ and C_2 with $1/(N(K-1)0.9)$. The factor 0.9 was set empirically. The formulation presented in Equation (10) constrained to (2), (3), (6), (7) and (8) in conjugation with the aggregation approach represented in Figure 3, results in our proposal titled *oTreeBagger*.

III. Experiments

Our experiments were conducted in sets of synthetic and real ordinal data, testing our method on the `syntheticI` and `syntheticII` for the synthetic datasets introduced in [6], and `LEV` real dataset [9]. `SyntheticI` dataset is uniformly distributed in the unit square $[0,1] \times [0,1] \subset \mathbb{R}^2$. For this dataset we assigned to each example $\mathbf{x}$ a class $y \in \{1,\ldots,5\}$ corresponding to $y = \min_{r \in \{1,2,3,4,5\}}\{r : b_{r-1} < 10(x_1 - 0.5)(x_2 - 0.5) + \varepsilon < b_r\}$ where $(b_0, b_1, b_2, b_3, b_4, b_5) = (-\infty, -1, -0.1, 0.25, 1, +\infty)$. ε is a random variable with normal distribution with zero mean and 0.125^2 of variance. `SyntheticII` data is uniformly distributed in the unit-circle, with the class y being assigned according to the radius of the point: $y = \left\lceil 3\sqrt{x_1^2 + x_2^2} \right\rceil$, $y \in \{1,\ldots,4\}$. For both datasets we generated 1000 example points. `LEV` dataset contains examples of anonymous lecturer evaluations, taken at the end of MBA courses and is composed by 4 features and 5 classes.

The baseline method (*TreeBagger*) used in our experiments consisted on the bagging approach with decision trees available in Matlab™ Statistical Toolbox. We opted to use the Gini index as splitting criterion.

[4] $\Delta = \Delta_1 \cup \Delta_2$ and $\Delta_1 \cap \Delta_2 = \emptyset$.

The grouping size M was evaluated from 1 to 5. The results presented in Figure 5 and Figure 6 show only the performance for a subset of these values for easier interpretation of the results. In these figures it is also clear the evolution of the learners throughout the increasing number of ensemble components. Due to the sensibility of these learners in regards to the number of training instances used, we conducted our experiments in 10%, 30% and 50% of training data. Our proposal outperformed the standard ensemble learner obtaining considerable gains in terms of performance. Logically, when the number of training instances increases this gain is more subtle, though.

IV. Conclusion

Learning on ordinal data has challenged many researchers to unfold the natural structure of the problem which, at the end, could lead to better performance results when compared with standard learning mechanisms. Despite the literature already presenting a rich collection in what concerns to this problem, there still exists a gap related to some classical methods. Decision trees are one example of it. Being well known and widely used within the machine learning community, as well the advantage of the interpretable capability, it is not straightforward its mapping towards ordinal data problems. In this work we proposed an improvement of [6] in order to reduce the over-regularised decision regions artifact through the usage of ensemble learning techniques. Results shown the benefits of our proposal in terms of accuracy gained when compared to a standard ensemble learning technique. Further studies will be taken to reduce the number of variables and constrains towards complexity diminution.

Acknowledgments

This work was partially supported by CNPq–Brazil through Program CNPq/Universidade do Porto/590008/2009-9 and conduced when Ricardo Sousa was in intership at Universidade Federal do Ceará. The first author would also like to thank to Luís Ferreira. This work was also partially funded by Fundação para a Ciência e a Tecnologia (FCT) - Portugal through project PTDC/SAU-ENB/114951/2009.

References

[1] R. Potharst and J. C. Bioch, "Quasi-monotone decision trees for ordinal classification," in *Proceedings of the Eighth Belgian-Dutch Conference on Machine Learning (BENELEARN'98)*, F. Verdenius and W. van den Broek, Eds., Agrotechnological Research Institute ATO-DLO, Wageningen, Netherlands, 1998, pp. 122–131.

[2] R. Potharst and J. Bioch, "A decision tree algorithm for ordinal classification," in *Advances in Intelligent Data Analysis*, ser. Lecture Notes in Computer Science, D. Hand, J. Kok, and M. Berthold, Eds. Springer Berlin / Heidelberg, 1999, vol. 1642, pp. 187–198.

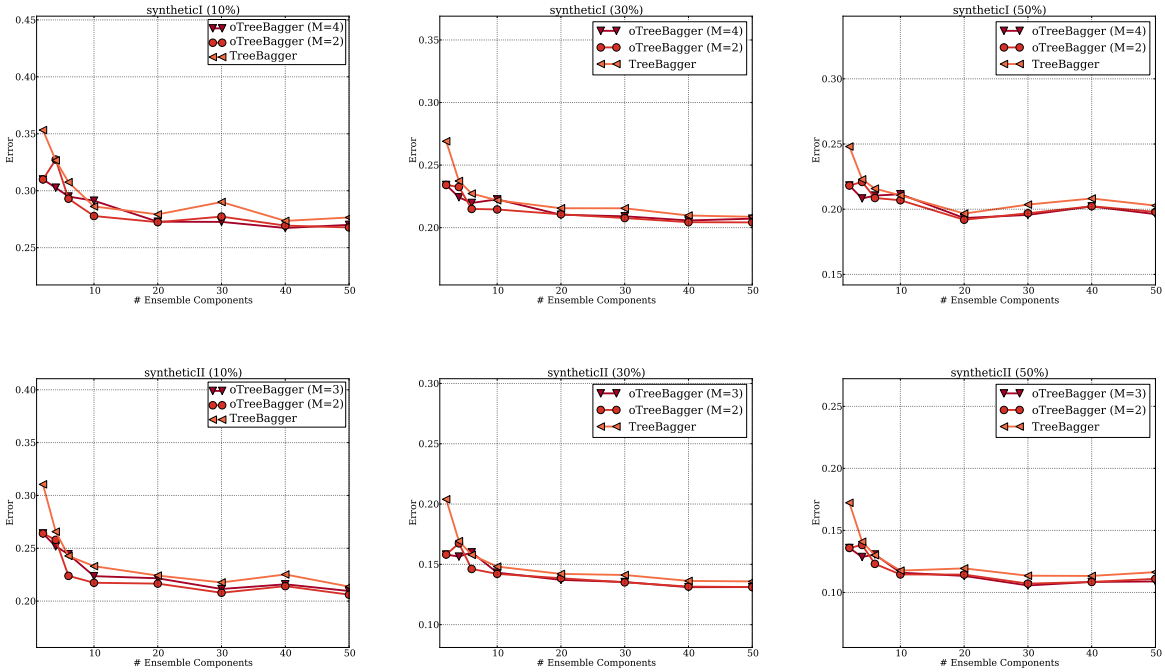


Figure 5: Results for synthetic datasets. Models trained with 10%, 30% and 50% of the 1000 instances in the left, centre and right plots, respectively.

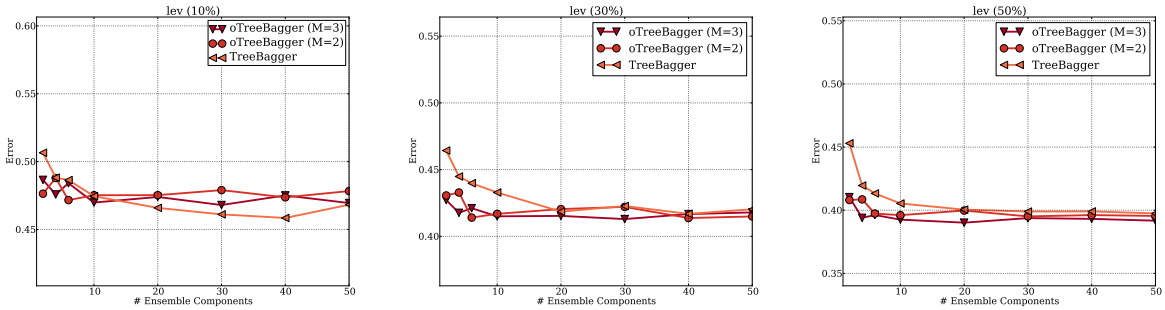


Figure 6: Results for real datasets. Models trained with 10%, 30% and 50% of the 1000 instances in the left, centre and right plots, respectively.

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

[4] S. Kramer, G. Widmer, B. Pfahringer, and M. d. Groeve, "Prediction of ordinal classes using regression trees," in *Proceedings of the 12th International Symposium on Foundations of Intelligent Systems*, ser. ISMIS '00. London, UK: Springer-Verlag, 2000, pp. 426–434.

[5] J. W. T. Lee and D.-Z. Liu, "Induction of ordinal decision trees," in *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, vol. 4, 2002, pp. 2220–2224.

[6] J. S. Cardoso and R. Sousa, "Classification models with global constraints for ordinal data," in *Proceedings of The Ninth International Conference on Machine Learning and Applications (ICMLA 2010)*, 2010.

[7] A. M. Zoubir and D. R. Iskander, "Bootstrap methods and applications," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 10–19, july 2007.

[8] C. Wang and J. Principe, "Training neural networks with additive noise in the desired signal," *Neural Networks, IEEE Transactions on*, vol. 10, no. 6, pp. 1511 –1517, nov 1999.

[9] A. Ben-David and L. Sterling, "Generating rules from examples of human multiattribute decision making should be simple," *Expert Systems with Applications*, vol. 31, no. 2, pp. 390 – 396, 2006.