Class switching ensembles for ordinal regression[⋆]

Pedro Antonio Gutiérrez[1⋆⋆], María Pérez-Ortiz[2], and Alberto Suárez[3]

[1] Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain. pagutierrez@uco.es

[2] Computer Science Department, Universidad Autónoma de Madrid, Madrid, Spain alberto.suarez@uam.es

[3] Department of Quantitative Methods, Universidad Loyola Andalucía, Córdoba, Spain. mariaperez@uloyola.es

Abstract. The term ordinal regression refers to classification tasks in which the categories have a natural ordering. The main premise of this learning paradigm is that the ordering can be exploited to generate more accurate predictors. The goal of this work is to design class switching ensembles that take into account such ordering so that they are more accurate in ordinal regression problems. In standard (nominal) class switching ensembles, diversity among the members of the ensemble is induced by injecting noise in the class labels of the training instances. Assuming that the classes are interchangeable, the the labels are modified at random. In ordinal class switching, the ordering between classes is taken into account by reducing the transition probabilities to classes that are further apart. In this manner smaller label perturbations in the ordinal scale are favoured. Two different specifications of these transition probabilities are considered; namely, an arithmetic and a geometric decrease with the absolute difference of the class ranks. These types of ordinal class switching ensembles are compared with an ensemble method that does not consider class-switching, a nominal class-switching ensemble, an ordinal variant of boosting, and two state-of-the-art ordinal classifiers based on support vector machines and Gaussian processes, respectively. These methods are evaluated and compared in a total of 15 datasets, using three different performance metrics. From the results of this evaluation one concludes that ordinal class-switching ensembles are more accurate than standard class-switching ones and than the ordinal ensemble method considered. Furthermore, their performance is comparable to the state-of-the-art ordinal regression methods considered in the analysis. Thus, class switching ensembles with specifically designed transition probabilities, which take into account the relationships between classes, are shown to provide very accurate predictions in ordinal regression problems.

Keywords: Class switching, ordinal regression, ensemble learning

[⋆] This work has been partially supported by projects TIN2014-54583-C2-1-R, TIN2013-42351-P, TIN2016-76406-P and TIN2015-70308-REDT of the Spanish Ministerial Commission of Science and Technology (MINECO, Spain) and FEDER funds (EU).

[⋆⋆] Corresponding author

1 Introduction

Ensemble methods have been successfully employed in numerous machine learning applications, including standard supervised learning problems [2,7,8], clustering [25] and image segmentation [13], among others. The goal in ensemble learning is to build a diverse collection of learners whose predictions are complementary. If the errors of the individual predictors are independent, they can be averaged in a combination step. This results in a global ensemble prediction that is more accurate than the one of individual classifiers. There are a number of strategies that can be used to build ensembles of diverse base learners. One of the most effective approaches to generate variability is to take advantage of the brittle character of the base learners and apply randomization techniques either in the dataset used for induction (e.g. training base learners with different bootstrap samples, as in bagging), or in the learning algorithm itself (e.g. build building random trees by considering splits only within a random subset of features, as in random forest). Another possibility is the injection of noise in the class labels. This technique was first introduced by Breiman under the name of *Output smearing* for regression, or *Output flipping* for classification problems [3]. In regression problems, the values of the dependent variable are contaminated with additive Gaussian noise. For classification, class labels are flipped at random with the restriction that the proportion of instances of the different classes is fixed. A direct extension of this technique, in which the class labels are simply modified at random, without ensuring that the class proportions are maintained, was analysed and seen to be more effective in ensembles of decision trees [19] and neural networks [18].

Despite their usefulness and demonstrated competitive performance, there are some areas of machine learning in which ensembles have been barely used. An example of this is the problem of ordinal regression, where these learning techniques have been applied only recently [9,15,16,21], with promising results. The problem of ordinal regression, also known in the literature as ordinal classification, is a supervised learning task in which the labels to be predicted are discrete, yet present an intrinsic ordering, which is relevant to the prediction problem at hand. For example, those surveys where students evaluate their teachers are usually based on an ordinal scale $\{poor, average, good, very\ good, excellent\}$. However, misclassifying *excellent* teachers as *poor* should be far more penalised than misclassifying them as *very good*. While it is possible to simply use standard classification techniques (which ignore the ordering of the labels) or regression techniques (which disregard the discrete nature of the labels and assume a specific distance between them), it is generally advantageous to consider specific methods that take into account the ordinal nature of the problem [12], not only for classification, but also in all the stages of the learning process, such as data preprocessing or performance evaluation. One of the most widely used approaches to ordinal regression are decomposition methods [12], which decompose the original ordinal variable into simpler classification problems [21,10], usually binary tasks, and the predictions of the corresponding classifiers are fused to produce an unique ordinal output. This approach can be seen as an ensemble where the diversity is introduced by the differences found in the classification tasks, and it is very natural in the case of ordinal regression because the classes can be joined according to different strategies (e.g. by a cascade binary utility model [14] or simply mixing neighbouring classes [10]). For example, for a given rank q, a direct question could be: "Is the label of pattern $\mathbf{x}$ greater than q?". Decomposition methods are popular within the ordinal classification literature, given that any binary classifier can be generally used as a base learner, without the need of reformulating the model to deal with the order of the classes. The main problem with this type of techniques is that the number of learners is relatively low (usually the number of classes minus one) and that fusing the different outputs is not straightforward [4]. Another differentiated group of ensemble-based approaches for ordinal regression are based on the concept of boosting. Some of these strategies rely on the confidence of a binary classifier [15,22,11], which can be used as an ordering preference, while others extend the well-known AdaBoost algorithm [16,23]. Finally, there are other strategies that make use of a base ordinal learner for the ensemble construction and introduce diversity as a term to be optimised during classifier construction [9] or that impose a global constraint to ensure that ordinal requirements are met [24].

In this article, we propose to design class-switching ensembles to address ordinal regression problems. To do so, different techniques are proposed to maintain and exploit the order information, based on a switching probability function that decreases with the rank difference. Our experiments compare seven different approaches in 15 datasets, showing that the ordinal class-switching approach outperforms the standard one and it is competitive with the state-of-the-art methods.

The rest of this paper is structured as follows: Section 2 discusses the design principles behind standard class-switching ensembles and describes how they are built. In section 3 the class switching method for ensemble generation is adapted to address ordinal regression problems. The effectiveness of the ensembles generated with the variants of class switching proposed is evaluated in an extensive set of experiments on benchmark ordinal regression problems, whose results are reported and analysed in section 4. Finally, section 5 summarises the contributions of this work and outlines some concluding remarks.

2 Class switching ensembles

In nominal classification problems, one generally assumes that the classes are independent of each other and their labels interchangeable. When designing machine learning algorithms, these properties are incorporated in the prediction mechanism. For instance, in decision trees the class label assignment is performed at the leaf nodes. Those instances that, as a result of the hierarchy of Boolean queries at the inner nodes of the decision tree, are assigned to a leaf node receive the label of the majority class of the training instances assigned to that node. Similarly, in ensembles, majority voting is used. In the voting process, each predictor is allowed to vote for a single class. In neural networks, 1-of-K

encoding is typically used for prediction problems with K classes. In this type of encoding, the kth class is represented as a vector of K bits, all of which are 0, except for the kth bit, which is equal to 1. The classes can be thought of as lying on the vertices of a regular simplex, and are therefore at equal distance from each other. The noise injection process in nominal class switching also assumes that classes are interchangeable: to build an individual ensemble classifier, a subset of training instances are selected at random. Then, the class label of these instances is modified also in a random fashion, assigning equal probabilities to switching the label to one of the other $K-1$ classes. Finally, a base learner is generated by applying the same base learning algorithm to the perturbed dataset. Once the ensemble has been completed, the predictions of the individual classifiers are combined by majority voting. Since the realizations are independent, the noise in the class labels is averaged out by the combination process. Furthermore, the variability induced can have a positive effect in the representation capacity of the ensemble.

In particular, when unpruned CART decision trees are used as base learners, class-switching ensembles achieve high prediction accuracy, provided that high switching rates (modifying the class label of a substantial fraction $\approx 0.6(K-1)/K$ of the training instances) and sufficiently large ensembles (of size ≥ 1000) are used. Similar improvements can be obtained using neural networks [18]. The goal of this work is to adapt this ensemble construction method to ordinal regression problems, in which class labels are ordered. If this ordering is relevant for prediction, designing a noise injection scheme that takes into account these relations among the class labels should lead to further accuracy improvements for this type of problems.

3 Ordinal class switching ensembles

Consider a labelled dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$. Assume that the dependent variable takes values in a finite set, $y_n \in \{c_1, c_2, \ldots, c_K\}$, which has an intrinsic ordering; that is, $c_1 < c_2 < \ldots < c_K$, where $<$ is an order relation provided by the nature of the classification problem.

To take advantage of the ordering relation among the class labels, we make the assumption that instances whose class labels are close to each other are more similar than instances whose class labels are further apart. Therefore, when selecting the modified class labels, one should assign higher probability to nearby classes. Specifically, we will choose a set of transition probabilities $\{p_{i\to j}; i,j = 1,\ldots,K\}$ with the following properties: If $|i-j| < |i-k| \Rightarrow p_{i\to j} < p_{i\to k}, \forall i,j,k \in \{1,\ldots,K\}$. Hence, the transition probability matrix should be V-shaped with respect to the class labels c_k, in the same way that cost matrices need to be V-shaped in the context of ordinal regression [17].

In this paper, we consider two different possibilities for constructing a V-shaped transition matrix:

– The probability of transition arithmetically decreases when the distance to the original class increases (arithmetic ordinal class switching, AOCS):

$$p^*_{i\to j} = \begin{cases} p^*, & \text{if} \quad i = j, \\ \dfrac{1-p^*}{|i-j|}, & \text{if} \quad i \neq j, \end{cases} \tag{1}$$

where $0 \leq p^* \leq 1$ is the parameter that sets the probability of not transitioning to a different class.

– The probability of transition geometrically decreases when the distance to the original class increases (geometric ordinal class switching, GOCS):

$$p^*_{i\to j} = \begin{cases} p^*, & \text{if} \quad i = j. \\ \dfrac{1-p^*}{2^{|i-j|}}, & \text{if} \quad i \neq j. \end{cases} \tag{2}$$

Given that the transition probabilities must add up to one, these matrices need to be normalised by rows:

$$p_{i\to j} = \frac{p^*_{i\to j}}{\sum_{k=1}^{K} p^*_{i\to k}}. \tag{3}$$

For example, for an ordinal regression problem with $K = 5$ classes and $p^* = 0.6$, the transition matrices are:

$$\mathbf{P}_{\text{AOCS}} = \begin{pmatrix} 0.54 & 0.18 & 0.12 & 0.09 & 0.07 \\ 0.16 & 0.49 & 0.16 & 0.11 & 0.08 \\ 0.11 & 0.16 & 0.47 & 0.16 & 0.11 \\ 0.08 & 0.11 & 0.16 & 0.49 & 0.16 \\ 0.07 & 0.09 & 0.12 & 0.18 & 0.54 \end{pmatrix}, \mathbf{P}_{\text{GOCS}} = \begin{pmatrix} 0.62 & 0.21 & 0.10 & 0.05 & 0.03 \\ 0.17 & 0.52 & 0.17 & 0.09 & 0.04 \\ 0.08 & 0.17 & 0.50 & 0.17 & 0.08 \\ 0.04 & 0.09 & 0.17 & 0.52 & 0.17 \\ 0.03 & 0.05 & 0.10 & 0.21 & 0.62 \end{pmatrix},$$

Note that, because of the normalization of class probabilities, the probability that a label remains unchanged (elements in the main diagonal of the matrix) decreases for central labels in the ordinal scale. This is sensible in ordinal regression, given that the changes in the extreme labels should be less likely.

4 Experiments

This section describes the experiments used to evaluate the performance of the ensembles generated with the variants of ordinal class switching proposed and an analysis of the results of these experiments.

4.1 Methods compared

The experiments are designed to compare the accuracy of ensembles generated with the two ordinal variants of class switching introduced in this paper (AOCS and GOCS, see Section 3 with standard (nominal) class switching (NCS, see Section 2) and an ensemble that does not make use of a label perturbation strategy for the different members of the ensemble (Orig). All the ensembles generated use classification trees as base learners, as in previous studies on class switching [19]. We used the implementation included in the Python `scikit-learn` machine learning framework [20], in which an optimised version of the CART algorithm is considered for tree induction. This implementation considers heuristic algorithms, where locally optimal decisions are made at each node. This makes the induction process non deterministic, therefore different trees can be obtained depending on the seed used for random number generation. This introduces diversity for the Orig algorithm, where no perturbation of the dataset is performed.

Besides evaluating whether ensembles generated with ordinal class switching methods are more accurate than their nominal counterparts, we compare their performance also with other state-of-the-art ordinal classifiers analysed in [12]; namely, we consider the reduction from ordinal regression to binary support vector machine classifiers (REDSVM) [17], the reformulation of Gaussian processes for ordinal regression (GPOR) [5] including automatic relevance determination, and an ensemble method, the ORBoost method with all margins [15].

4.2 Measures of performance

Different metrics can be used to evaluate ordinal regression classifiers. The most common ones are accuracy (Acc) and Mean Absolute Error (MAE). Acc is the rate of correctly classified patterns:

$$Acc = \frac{1}{N}\sum_{i=1}^{N} I(y_i^* = y_i),$$

where y_i is the correct label of the ith instance, y_i^* is the predicted one, and $I(\cdot)$ is a Boolean test. This measure characterises the global performance in the classification task, without taking into account the ordering of the classes.

The Mean Absolute Error is an average deviation in absolute value of the predicted rank from the true one [1]:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|,$$

where $\mathcal{O}$, $\mathcal{O}(c_j) = j, 1 \leq j \leq K$. MAE values range from 0 to $K-1$ (maximum deviation in number of ranks between two labels). Given that some of the datasets considered are imbalanced, we also consider the average of the MAEs across classes ($AMAE$) [1]:

$$AMAE = \frac{1}{K}\sum_{j=1}^{K} MAE_j = \frac{1}{K}\sum_{j=1}^{K}\frac{1}{n_j}\sum_{i=1}^{n_j} |\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|,$$

where n_j is the number of patterns in class j. The value of $AMAE$ range from 0 to $K-1$.

Table 1: Characteristics of the benchmark datasets used in the experiments.

Dataset	#Pat.	#Attr.	#Classes	Class distribution
ERA	1000	4	9	$(92, 142, 181, 172, 158, 118, 88, 31, 18)$
ESL	488	4	9	$(2, 12, 38, 100, 116, 135, 62, 19, 4)$
LEV	1000	4	5	$(93, 280, 403, 197, 27)$
SWD	1000	10	4	$(32, 352, 399, 217)$
automobile	205	71	6	$(3, 22, 67, 54, 32, 27)$
balance-scale	625	4	3	$(288, 49, 288)$
bondrate	57	37	5	$(6, 33, 12, 5, 1)$
eucalyptus	736	91	5	$(180, 107, 130, 214, 105)$
newthyroid	215	5	3	$(30, 150, 35)$
pasture	36	25	3	$(12, 12, 12)$
squash-stored	52	51	3	$(23, 21, 8)$
squash-unstored	52	52	3	$(24, 24, 4)$
tae	151	54	3	$(49, 50, 52)$
toy	300	2	5	$(35, 87, 79, 68, 31)$
winequality-red	1599	11	6	$(10, 53, 681, 638, 199, 18)$

4.3 Datasets and experimental setup

A battery of 15 ordinal regression datasets is used for evaluation. Their characteristics are summarised in Table 1. The selected datasets are very different in terms of numbers of patterns, attributes, classes and class distribution to ensure that the conclusions of the study cover a sufficiently wide rage of ordinal regression problems. The experimental protocol is similar to the one used in [12]. The results reported are averages over 30 random partitions into training (3/4 of the data) and test sets (1/4 of the data), considering the same partitions than in [12][4]. Consequently, the results for GPOR, ORBoost and REDSVM were directly taken from [12].

In all cases (Orig, NCS, AOCS and GOCS) ensembles of $T = 1001$ predictors are used. The outputs of the ensemble classifiers are combined using a soft voting rule: The global ensemble prediction for a given instance, characterised by the vector of attributes $\mathbf{x}$, is $c_j = \arg\max_{c_i} \sum_{t=1}^{T} p_{ti}(\mathbf{x})$, where $p_{ti}(\mathbf{x})$ is the probability assigned by the tth ensemble classifier to the class label c_i. These probabilities are approximated as the fraction of samples of the corresponding class in the leaf node to which the instance is assigned. In class switching, after preliminary experiments, p^* is set to 0.6.

The hyperparameters for REDSVM are selected by using a nested five fold cross-validation over the training set. The criterion used to determine the optimal hyperparameter values is MAE. A Gaussian kernel function is used, and both the value of the cost parameter C and the width of the kernel are determined considering the range $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. In GPOR, the hyperparameters are determined by part of the optimisation process. Following the recommendation given in [15], the ensemble size in ORBoost is $T = 2000$. In this method, nor-

[4] Available at `http://www.uco.es/grupos/ayrna/orreview`

malised sigmoid functions are used as base learners. The smoothness parameter, γ is set to 4.

4.4 Results

In Table 2 the values of Acc, MAE and $AMAE$ obtained by the different methods in the test set are presented. The measures are averages and standard deviations (in a smaller font) over the 30 random training/test partitions.

In Table 3 the average ranks (in terms of the values of Acc, MAE and $AMAE$ in the test partitions for the 15 datasets) are presented. Specifically, rank 1 corresponds to the best performance and rank 7 to the worst one. From the results presented in this table, it is apparent that the *Orig* ensembles have poor performance results. This is probably related to the low diversity among the ensemble classifiers: since the same unperturbed training data are used to build the individual classifiers, the only source of variability is the intrinsic randomness of the base learning algorithm. Both ordinal class switching ensembles (AOCS and GOCS) yield better performance than standard (nominal) class switching (NCS). The geometric decrease (GOCS) leads to better overall results than the arithmetic one (AOCS) when the performance is measured in terms of measures such as Acc and MAE. However AOCS outperforms GOCS when the metric is class-specific (e.g. using $AMAE$). In general, the accuracy is competitive with respect to the state-of-the-art methods considered (GPOR and REDSVM). The ordinal class-switching ensembles are clearly more accurate than ORBoost ensembles. The statistical significance of the differences of performance between the different methods is determined using the the guidelines given in [6]. Specifically, a non-parametric Friedman's test (at a significance level of $\alpha = 0.10$) has been applied for Acc, MAE and $AMAE$ rankings. The confidence interval is, in this case, $C_0 = (0, F_{(\alpha=0.10)} = 1.85)$. The values of the statistic are F_{Acc}: $2.53 \notin C_0$, F_{MAE}: $3.34 \notin C_0$ and F_{AMAE}: $2.17 \notin C_0$. Consequently, the test rejects the null-hypothesis that, as measured by the average rank, the algorithms have similar performance.

Considering AOCS and GOCS as the control methods, we apply the post-hoc Holm's test [6]. The performance of the i-th and j-th algorithms are are compared using the statistic:

$$z = \frac{R_i - R_j}{\sqrt{\frac{J(J+1)}{6N}}},$$

where J is the number of algorithms ($J = 7$, in our case), N is the number of datasets ($N = 15$, in our case) and R_i is the average rank of the i-th method (see Table 3). Asymptotically, this statistic is normally distributed, which allows us to quantify the significance of the differences observed at the corresponding significance level (α). The value of α is adjusted to take into account that multiple comparisons are made. The adjustment is made by a a sequential procedure: the ordered p-values are $p_1, p_2, \ldots, p_{J-1}$, so that $p_1 \leq p_2 \leq \ldots \leq p_{J-1}$, and each p_i is compared with $\alpha/(J-i)$. The results of these tests are presented in Table 4.

Table 2: Average and standard deviation of *Acc*, *MAE* and *AMAE* values obtained for the different methods compared

	Acc						
Dataset	Orig	NCS	AOCS	GOCS	GPOR	ORBoost	REDSVM
ERA	$0.257_{0.024}$	$0.255_{0.025}$	$\mathit{0.259_{0.029}}$	$0.258_{0.028}$	$\mathbf{0.288_{0.027}}$	$0.240_{0.021}$	$0.249_{0.019}$
ESL	$0.642_{0.033}$	$0.651_{0.035}$	$0.652_{0.035}$	$0.653_{0.035}$	$\mathbf{0.713_{0.031}}$	$0.677_{0.022}$	$\mathit{0.713_{0.030}}$
LEV	$0.614_{0.021}$	$0.613_{0.024}$	$\mathit{0.621_{0.023}}$	$0.621_{0.022}$	$0.612_{0.030}$	$0.609_{0.029}$	$\mathbf{0.627_{0.024}}$
SWD	$0.540_{0.029}$	$0.551_{0.026}$	$0.562_{0.025}$	$0.562_{0.024}$	$\mathbf{0.578_{0.031}}$	$0.561_{0.032}$	$\mathit{0.571_{0.027}}$
automobile	$0.788_{0.061}$	$\mathbf{0.820_{0.047}}$	$0.816_{0.049}$	$\mathit{0.817_{0.050}}$	$0.611_{0.073}$	$0.706_{0.055}$	$0.683_{0.070}$
balance-scale	$0.772_{0.019}$	$0.802_{0.023}$	$0.775_{0.024}$	$0.775_{0.024}$	$0.966_{0.012}$	$\mathit{0.968_{0.016}}$	$\mathbf{0.999_{0.004}}$
bondrate	$0.463_{0.110}$	$0.524_{0.074}$	$0.553_{0.070}$	$0.559_{0.070}$	$\mathbf{0.578_{0.032}}$	$0.542_{0.091}$	$\mathit{0.564_{0.055}}$
eucalyptus	$0.609_{0.030}$	$0.674_{0.032}$	$\mathit{0.681_{0.032}}$	$0.679_{0.032}$	$\mathbf{0.686_{0.034}}$	$0.620_{0.029}$	$0.638_{0.035}$
newthyroid	$0.940_{0.038}$	$\mathbf{0.969_{0.017}}$	$\mathit{0.969_{0.018}}$	$\mathit{0.969_{0.018}}$	$0.966_{0.024}$	$0.958_{0.029}$	$0.968_{0.023}$
pasture	$\mathit{0.762_{0.105}}$	$\mathbf{0.781_{0.147}}$	$0.753_{0.138}$	$0.753_{0.138}$	$0.522_{0.178}$	$0.700_{0.121}$	$0.674_{0.116}$
squash-stored	$0.624_{0.119}$	$\mathit{0.699_{0.110}}$	$\mathbf{0.704_{0.101}}$	$\mathbf{0.704_{0.101}}$	$0.451_{0.100}$	$0.636_{0.124}$	$0.621_{0.132}$
squash-unstored	$0.764_{0.112}$	$\mathit{0.835_{0.085}}$	$\mathbf{0.839_{0.085}}$	$\mathbf{0.839_{0.085}}$	$0.644_{0.162}$	$0.703_{0.098}$	$0.731_{0.119}$
tae	$0.582_{0.057}$	$0.570_{0.071}$	$0.580_{0.062}$	$0.580_{0.062}$	$0.328_{0.041}$	$\mathit{0.597_{0.057}}$	$\mathbf{0.601_{0.068}}$
toy	$0.885_{0.033}$	$0.936_{0.025}$	$0.934_{0.026}$	$0.933_{0.026}$	$\mathit{0.954_{0.022}}$	$0.948_{0.025}$	$\mathbf{0.977_{0.012}}$
winequality-red	$0.615_{0.020}$	$0.685_{0.017}$	$\mathit{0.685_{0.016}}$	$\mathbf{0.686_{0.016}}$	$0.606_{0.015}$	$0.666_{0.021}$	$0.627_{0.020}$
	MAE						
Dataset	Orig	NCS	AOCS	GOCS	GPOR	ORBoost	REDSVM
ERA	$1.383_{0.065}$	$1.372_{0.065}$	$1.335_{0.060}$	$1.334_{0.061}$	$\mathit{1.241_{0.051}}$	$1.250_{0.041}$	$\mathbf{1.219_{0.044}}$
ESL	$0.385_{0.040}$	$0.370_{0.042}$	$0.370_{0.040}$	$0.367_{0.040}$	$\mathbf{0.301_{0.035}}$	$0.340_{0.025}$	$\mathit{0.306_{0.037}}$
LEV	$0.425_{0.027}$	$0.429_{0.031}$	$\mathit{0.417_{0.027}}$	$0.417_{0.026}$	$0.422_{0.031}$	$0.434_{0.030}$	$\mathbf{0.410_{0.023}}$
SWD	$0.499_{0.034}$	$0.484_{0.031}$	$0.467_{0.028}$	$0.466_{0.027}$	$\mathbf{0.440_{0.032}}$	$0.463_{0.036}$	$\mathit{0.445_{0.031}}$
automobile	$0.327_{0.108}$	$\mathbf{0.263_{0.074}}$	$0.266_{0.074}$	$\mathit{0.265_{0.076}}$	$0.594_{0.131}$	$0.348_{0.079}$	$0.403_{0.092}$
balance-scale	$0.264_{0.028}$	$0.230_{0.031}$	$0.250_{0.031}$	$0.250_{0.031}$	$0.034_{0.012}$	$\mathit{0.032_{0.017}}$	$\mathbf{0.001_{0.004}}$
bondrate	$0.739_{0.163}$	$0.630_{0.104}$	$0.592_{0.105}$	$\mathit{0.587_{0.103}}$	$0.624_{0.062}$	$\mathbf{0.531_{0.110}}$	$0.613_{0.081}$
eucalyptus	$0.447_{0.035}$	$0.350_{0.036}$	$\mathit{0.343_{0.036}}$	$0.344_{0.037}$	$\mathbf{0.331_{0.038}}$	$0.415_{0.036}$	$0.395_{0.036}$
newthyroid	$0.060_{0.038}$	$0.031_{0.017}$	$\mathit{0.031_{0.018}}$	$\mathit{0.031_{0.018}}$	$0.034_{0.024}$	$0.042_{0.029}$	$\mathbf{0.029_{0.022}}$
pasture	$\mathit{0.238_{0.105}}$	$\mathbf{0.219_{0.147}}$	$0.247_{0.138}$	$0.247_{0.138}$	$0.489_{0.190}$	$0.300_{0.121}$	$0.330_{0.111}$
squash-stored	$0.424_{0.156}$	$\mathit{0.335_{0.132}}$	$\mathbf{0.327_{0.122}}$	$\mathbf{0.327_{0.122}}$	$0.626_{0.148}$	$0.364_{0.124}$	$0.346_{0.145}$
squash-unstored	$0.236_{0.112}$	$\mathit{0.168_{0.086}}$	$\mathbf{0.161_{0.085}}$	$\mathbf{0.161_{0.085}}$	$0.356_{0.162}$	$0.305_{0.106}$	$0.262_{0.122}$
tae	$0.521_{0.092}$	$0.553_{0.104}$	$0.523_{0.095}$	$0.523_{0.095}$	$0.861_{0.155}$	$\mathit{0.504_{0.088}}$	$\mathbf{0.461_{0.060}}$
toy	$0.117_{0.035}$	$0.064_{0.025}$	$0.066_{0.026}$	$0.067_{0.026}$	$\mathit{0.046_{0.022}}$	$0.052_{0.025}$	$\mathbf{0.024_{0.013}}$
winequality-red	$0.457_{0.026}$	$0.351_{0.019}$	$\mathit{0.349_{0.019}}$	$\mathbf{0.348_{0.019}}$	$0.425_{0.017}$	$0.365_{0.021}$	$0.417_{0.020}$
	AMAE						
Dataset	Orig	NCS	AOCS	GOCS	GPOR	ORBoost	REDSVM
ERA	$1.428_{0.122}$	$1.405_{0.114}$	$\mathbf{1.370_{0.099}}$	$\mathit{1.372_{0.102}}$	$1.372_{0.108}$	$1.427_{0.102}$	$1.413_{0.086}$
ESL	$0.586_{0.086}$	$0.572_{0.081}$	$0.569_{0.084}$	$0.569_{0.083}$	$\mathit{0.477_{0.094}}$	$0.554_{0.096}$	$\mathbf{0.459_{0.116}}$
LEV	$0.603_{0.061}$	$0.603_{0.054}$	$\mathbf{0.601_{0.051}}$	$\mathit{0.602_{0.051}}$	$0.654_{0.050}$	$0.615_{0.044}$	$0.620_{0.046}$
SWD	$0.579_{0.052}$	$0.581_{0.054}$	$\mathit{0.579_{0.052}}$	$0.580_{0.051}$	$0.589_{0.040}$	$\mathbf{0.576_{0.039}}$	$0.608_{0.034}$
automobile	$0.316_{0.131}$	$0.320_{0.127}$	$\mathit{0.314_{0.120}}$	$\mathbf{0.313_{0.120}}$	$0.792_{0.200}$	$0.400_{0.120}$	$0.464_{0.135}$
balance-scale	$0.463_{0.022}$	$0.440_{0.022}$	$0.455_{0.023}$	$0.455_{0.023}$	$0.051_{0.023}$	$\mathit{0.042_{0.029}}$	$\mathbf{0.001_{0.003}}$
bondrate	$1.167_{0.320}$	$1.204_{0.182}$	$1.161_{0.168}$	$1.161_{0.163}$	$1.360_{0.122}$	$\mathbf{0.839_{0.260}}$	$\mathit{1.144_{0.275}}$
eucalyptus	$0.470_{0.039}$	$0.378_{0.040}$	$\mathit{0.370_{0.038}}$	$0.372_{0.039}$	$\mathbf{0.362_{0.040}}$	$0.434_{0.040}$	$0.429_{0.039}$
newthyroid	$0.085_{0.056}$	$\mathit{0.056_{0.035}}$	$0.061_{0.039}$	$0.061_{0.039}$	$0.062_{0.049}$	$0.067_{0.049}$	$\mathbf{0.048_{0.040}}$
pasture	$\mathit{0.238_{0.105}}$	$\mathbf{0.219_{0.147}}$	$0.247_{0.138}$	$0.247_{0.138}$	$0.489_{0.190}$	$0.300_{0.121}$	$0.330_{0.111}$
squash-stored	$0.452_{0.224}$	$0.392_{0.197}$	$0.392_{0.192}$	$0.392_{0.192}$	$0.797_{0.234}$	$\mathbf{0.368_{0.129}}$	$\mathit{0.386_{0.162}}$
squash-unstored	$0.207_{0.135}$	$\mathbf{0.170_{0.137}}$	$\mathit{0.172_{0.143}}$	$\mathit{0.172_{0.143}}$	$0.443_{0.226}$	$0.341_{0.174}$	$0.383_{0.184}$
tae	$0.521_{0.092}$	$0.553_{0.104}$	$0.522_{0.095}$	$0.522_{0.095}$	$0.863_{0.164}$	$\mathit{0.503_{0.087}}$	$\mathbf{0.459_{0.059}}$
toy	$0.118_{0.035}$	$0.063_{0.023}$	$0.068_{0.026}$	$0.071_{0.028}$	$\mathit{0.044_{0.025}}$	$0.054_{0.027}$	$\mathbf{0.024_{0.016}}$
winequality-red	$0.971_{0.118}$	$0.958_{0.076}$	$\mathbf{0.952_{0.076}}$	$\mathit{0.956_{0.078}}$	$1.065_{0.065}$	$0.974_{0.048}$	$1.080_{0.083}$

For each dataset the best result is marked in **bold** face; the second best in *italics*

The tests conclude that the differences between AOCS and Orig are statistically significant for *Acc* and *MAE*, as well as those between GOCS and Orig for the same metrics. In terms of *AMAE*, AOCS is significantly better than GPOR and Orig, while GOCS is significantly better than GPOR.

Table 3: Average test rankings over the 15 datasets for *Acc*, *MAE* and *AMAE*.

Method	Orig	NCS	AOCS	GOCS	GPOR	ORBoost	REDSVM
Acc	5.40	4.00	*3.13*	**3.07**	4.20	4.67	3.53
MAE	5.80	4.20	3.40	*3.20*	4.33	4.07	**3.00**
AMAE	5.00	3.93	**3.00**	*3.40*	5.13	3.73	3.80

The best result is in bold face and the second one in italics

Table 4: Results of the Holm procedure using AOCS and GOCS as control methods: corrected α values, compared method and p-values, ordered by significance.

		Acc			
		Control Method			
		AOCS		GOCS	
i	$\alpha/(7-i)$	Method	p-value	Method	p-value
1	0.01667	Orig	0.00406_+	Orig	0.00310_+
2	0.02000	ORBoost	0.05191	ORBoost	0.04252
3	0.02500	GPOR	0.17630	GPOR	0.15079
4	0.03333	NCS	0.27190	NCS	0.23673
5	0.05000	REDSVM	0.61209	REDSVM	0.55412
6	0.10000	GOCS	0.93265	AOCS	0.93265
		MAE			
		Control Method			
		AOCS		GOCS	
i	$\alpha/(7-i)$	Method	p-value	Method	p-value
1	0.01667	Orig	0.00235_+	Orig	0.00098_+
2	0.02000	GPOR	0.23673	GPOR	0.15079
3	0.02500	NCS	0.31049	NCS	0.20489
4	0.03333	ORBoost	0.39802	ORBoost	0.27190
5	0.05000	REDSVM	0.61209	REDSVM	0.79985
6	0.10000	GOCS	0.79985	AOCS	0.79985
		AMAE			
		Control Method			
		AOCS		GOCS	
i	$\alpha/(7-i)$	Method	p-value	Method	p-value
1	0.01667	GPOR	0.00684_+	GPOR	0.02799_+
2	0.02000	Orig	0.01123_+	Orig	0.04252
3	0.02500	NCS	0.23673	NCS	0.49896
4	0.03333	REDSVM	0.31049	AOCS	0.61209
5	0.05000	ORBoost	0.35254	REDSVM	0.61209
6	0.10000	GOCS	0.61209	ORBoost	0.67261

$+$: statistical difference with $\alpha = 0.10$

5 Conclusions

In this work, we propose to take into account the ordering between classes to improve the accuracy of class switching ensembles in ordinal regression problems.

To this end, the class switching protocol is modified so that, in the noise injection process, a class label that is closer to the original one has a higher probability of being selected. Two variants of the class switching procedure are considered. In the first one (AOCS) the probability of changing original label to another one decreases arithmetically with the distance between the target label and the original one. In the second one, the decrease with this distance is geometric (GOCS). For global measures of performance, such as accuracy (*Acc*) or the Mean Average Error (MAE), the geometric scheme provides the best results. However, when measures that incorporate more information on the relationship among the classes, such as the average of the MAEs across classes (AMAE), are used, the arithmetic scheme performs better than the geometric scheme. Both types of ordinal class switching ensembles are more accurate than standard (nominal) class switching, which assumes that the classes are interchangeable (i.e., it discards the information on the ordering of the classes). Furthermore, their generalization capacity is comparable to, and in some cases better, the state-of-the-art ordinal regression methods.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA'09). pp. 283–287 (2009)
2. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36(1), 105–139 (1999), http://dx.doi.org/10.1023/A:1007515423169
3. Breiman, L.: Randomizing outputs to increase prediction accuracy. Machine Learning 40(3), 229–242 (2000), http://dx.doi.org/10.1023/A:1007682208299
4. Cardoso, J.S., da Costa, J.F.P.: Learning to classify ordinal data: the data replication method. Journal of Machine Learning Research 8, 1393–1429 (2007)
5. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. J. of Machine Learning Research 6, 1019–1041 (2005)
6. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
7. Dietterich, T.G.: Ensemble methods in machine learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems. pp. 1–15. MCS '00, Springer-Verlag, London, UK, UK (2000)
8. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research 15, 3133–3181 (2014), http://jmlr.org/papers/v15/delgado14a.html
9. Fernandez-Navarro, F., Gutiérrez, P.A., Hervás-Martínez, C., Yao, X.: Negative Correlation Ensemble Learning for Ordinal Regression. IEEE Transactions on Neural Networks and Learning Systems 24(11), 1836–1849 (November 2013), http://dx.doi.org/10.1109/TNNLS.2013.2268279, jCR (2013): 4.370 (category COMPUTER SCIENCE, THEORY &amp; METHODS, position 2/102 Q1)
10. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Proceedings of the 12th European Conference on Machine Learning. pp. 145–156. EMCL '01, Springer-Verlag, London, UK, UK (2001)

11. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. J. Mach. Learn. Res. 4, 933–969 (Dec 2003), `http://dl.acm.org/citation.cfm?id=945365.964285`
12. Gutiérrez, P.A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., Hervas-Martinez, C.: Ordinal regression methods: survey and experimental study. IEEE Transactions on Knowledge and Data Engineering 28(1), 127–146 (2016)
13. Kim, H., Thiagarajan, J.J., Bremer, P.T.: Image segmentation using consensus from hierarchical segmentation ensembles. In: ICIP. pp. 3272–3276. IEEE (2014)
14. Kwon, Y.S., Han, I., Lee, K.C.: Ordinal pairwise partitioning (opp) approach to neural networks training in bond rating. Int. Syst. in Accounting, Finance and Management 6(1), 23–40 (1997)
15. Lin, H.T., Li, L.: Large-margin thresholded ensembles for ordinal regression: Theory and practice. In: Balczar, J.L., Long, P.M., Stephan, F. (eds.) Proc. of the 17th Algorithmic Learning Theory International Conference. Lecture Notes in Artificial Intelligence (LNAI), vol. 4264, pp. 319–333. Springer-Verlag (October 2006)
16. Lin, H.T., Li, L.: Combining ordinal preferences by boosting. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD). pp. 69–83 (2009)
17. Lin, H.T., Li, L.: Reduction from cost-sensitive ordinal ranking to weighted binary classification. Neural Comput. 24(5), 1329–1367 (2012)
18. Martínez-Muñoz, G., Sánchez-Martínez, A., Hernández-Lobato, D., Suárez, A.: Building ensembles of neural networks with class-switching. In: Artificial Neural Networks - ICANN 2006. pp. 178–187 (2006)
19. Martínez-Muñoz, G., Suárez, A.: Switching class labels to generate classification ensembles. Pattern Recognition 38(10), 1483–1494 (2005)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12(Oct), 2825–2830 (2011)
21. Pérez-Ortiz, M., Gutiérrez, P.A., Hervás-Martínez, C.: Projection based ensemble learning for ordinal regression. IEEE Transactions on Cybernetics 44(5), 681–694 (May 2014), `http://www.uco.es/grupos/ayrna/elor2013`
22. Rennie, J.D.M.: Loss functions for preference levels: Regression with discrete ordered labels. In: Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling. pp. 180–186 (2005)
23. Riccardi, A., Fernandez-Navarro, F., Carloni, S.: Cost-Sensitive AdaBoost Algorithm for Ordinal Regression Based on Extreme Learning Machine. IEEE Transaction on Cybernetics 44(10), 1898–1909 (2014)
24. Sousa, R., Cardoso, J.S.: Ensemble of decision trees with global constraints for ordinal classification. In: 2011 11th International Conference on Intelligent Systems Design and Applications. pp. 1164–1169 (Nov 2011)
25. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)