Ordinal regression methods: survey and experimental study

Pedro Antonio Gutiérrez, *Member, IEEE,* María Pérez-Ortiz, Javier Sánchez-Monedero, Francisco Fernández-Navarro, and César Hervás-Martínez, *Senior Member, IEEE*

Abstract—Ordinal regression problems are those machine learning problems where the objective is to classify patterns using a categorical scale which shows a natural order between the labels. Many real-world applications present this labelling structure and that has increased the number of methods and algorithms developed over the last years in this field. Although ordinal regression can be faced using standard nominal classification techniques, there are several algorithms which can specifically benefit from the ordering information. Therefore, this paper is aimed at reviewing the state of the art on these techniques and proposing a taxonomy based on how the models are constructed to take the order into account. Furthermore, a thorough experimental study is proposed to check if the use of the order information improves the performance of the models obtained, considering the most significant published approaches within the taxonomy. The results confirm that ordering information benefits ordinal models improving their accuracy and the closeness of the predictions to actual targets in the ordinal scale.

Index Terms—Ordinal regression, ordinal classification, binary decomposition, threshold methods, augmented binary classification, proportional odds model, support vector machines, discriminant learning, artificial neural networks

1 Introduction

Learning to classify or to predict numerical values from prelabelled patterns is one of the central research topics in machine learning and data mining [1]–[3]. However, less attention has been paid to ordinal regression (also called ordinal classification) problems, where the labels of the target variable exhibit a natural ordering. For example, student satisfaction surveys usually involve rating teachers based on an ordinal scale $\{poor, average, good, very\ \ good, excellent\}$. Hence, class labels are imbued with order information, e.g. a sample vector associated with class label $average$ has a higher rating (or better) than another from the $poor$ class, but $good$ class is better than both. When dealing with this kind of problems, two facts are decisive: misclassification costs are not the same for different errors (it is clear that misclassifying an $excellent$ teacher as $poor$ should be more penalised than misclassifying him/her as $very\ good$) and the ordering information can be used to construct more accurate models. A further distinction is made by Anderson [4], which differentiates two major types of ordinal categorical variables, "grouped continuous variables" and "assessed ordered categorical variables". The first one is a discretised version of an underlying continuous variable, which could be observed itself. The second one covers those variables where a user provides his/her judgement on the grade of the ordered categorical variable. However, imposing an ordering is meaningful for both cases.

Ordinal regression problems are very common in many research areas, and they have been frequently considered as standard nominal problems which can lead to non-optimal solutions. Indeed, ordinal regression problems can be said to be between classification and regression, presenting some similarities and differences. Some of the fields where ordinal regression is found are medical research [5]–[11], age estimation [12], brain computer interface [13], credit rating [14]–[17], econometric modelling [18], face recognition [19]–[21], facial beauty assessment [22], image classification [23], wind speed prediction [24], social sciences [25], text classification [26], and more. All these works are examples of application of specifically designed ordinal regression models, where the ordering consideration improves their performance with respect to their nominal counterparts.

In statistics, ordinal data were firstly studied by using a link function able to model the underlying probability for generating ordinal labels [4]. The field of ordinal regression has evolved in the last decade, with a plethora of noteworthy research progress made in supervised learning [27], from support vector machine (SVM) formulations [28], [29] to Gaussian processes [30] or discriminant learning [31], to name a few. However, up to the authors' knowledge, these methods have not yet been categorised in a general taxonomy, which is essential for further research and for identifying the developments made and the present state of existing methods. This paper contributes a review of the state-of-the-art of ordinal regression, a taxonomy proposal to

This work has been partially subsidised by the TIN2011-22794 project of the Spanish Ministry of Economy and Competitiveness (MINECO), FEDER funds and the P2011-TIC-7508 project of the "Junta de Andalucía" (Spain).
P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero and C. Hervás-Martínez are with the Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, Albert Einstein building, 14017 - Córdoba, Spain, e-mail:{pagutierrez,i82perom,jsanchezm,chervas}@uco.es
F. Fernández-Navarro is with the Department of Mathematics and Engineering, Loyola University Andalucía, Spain, e-mail: i22fenaf@uco.es

better organise the advances in this field, and an experimental study with a complete repository of datasets and a total of 16 ordinal regression methods (including a software tool to run and test all the methods).

Several objectives motivate the experimental study. First of all, our focus is on evaluating the necessity of taking ordering information into account. In [32], ordinal meta-models were compared with respect to their nominal counterparts to check their ability to exploit ordinal information. The work concludes that such meta-methods do exploit ordinal information and may yield better performance. However, as will be analysed in this work, specifically designed ordinal regression methods can further improve the results with respect to meta-model approaches. Another study [33] argues that ordinal classifiers may not present meaningful advantages over the analogue non-ordinal methods, based on accuracy and Cohen's Kappa statistic [34]. The results of the present review show that statistically significant differences are found when using measures which take the order into account, which is the case of the Mean Absolute Error (MAE), i.e. the average deviation between predicted and actual targets in number of categories. The second main motivation of this paper is to provide some guidelines to decide on the best methods in terms of accuracy, MAE and computational time. Since there are not specific repositories of ordinal regression datasets, proposals are usually evaluated using discretised regression ones, where the target variable is simply divided into different bins or classes. 24 of these discretised datasets are used for our study, in addition to 17 real benchmark ordinal regression datasets extracted from public repositories. The last objective is to evaluate whether the methods behave differently depending on the nature of the datasets.

This paper is a significant extension of a preliminary conference version [35]: a deeper analysis of the state-of-the-art has been performed, including most recent proposals and a taxonomy to group them. Moreover, the experimental study includes more methods and datasets. The rest of the paper is organised as follows. Section 2 introduces the problem of ordinal regression and briefly describes its differences from some related machine learning topics outside the scope of this paper. Section 3 revises ordinal regression state-of-the-art by grouping different methods with a proposed taxonomy. The main representatives of each family are then empirically compared in Section 4, where the experiments are described and the corresponding results are studied and discussed. Finally, Section 5 deals with the main achievements.

2 Notation and Nature of the Problem

2.1 Problem definition

The ordinal regression problem consists on predicting the label y of an input vector $\mathbf{x}$, where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$ and $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$, i.e. $\mathbf{x}$ is in a K-dimensional input space and y is in a label space of Q different labels. These labels form categories or groups of patterns, and the objective is to find a classification rule or function $f : \mathcal{X} \to \mathcal{Y}$ to predict the categories of new patterns, given a training set of N points, $D = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$. A natural label ordering is included for ordinal regression, $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \ldots \prec \mathcal{C}_Q$, where $\prec$ is an order relation given by the nature of the classification problem. Many ordinal regression measures and algorithms consider the rank of the label, i.e. the position of the label in the ordinal scale, which can be expressed by the function $\mathcal{O}(\cdot)$, in such a way that $\mathcal{O}(\mathcal{C}_q) = q, q = 1, \ldots, Q$. The difference between this setting and other related ones is now established. The assumption of an order between class labels makes that two different elements of $\mathcal{Y}$ can be always compared by using the relation $\prec$, which is not possible under the nominal classification setting. If compared to regression (where $y \in \mathbb{R}$), it is true that real values in $\mathbb{R}$ can be ordered by the standard $<$ operator, but labels in ordinal regression ($y \in \mathcal{Y}$) do not carry metric information, so the category serves as a qualitative indication of the pattern rather than a quantitative one.

2.2 Ordinal regression in the context of ranking and sorting

Although ordinal regression has been paid attention recently, the amount of related research topics is worth to be mentioned. First, it is important to remark the differences between ordinal regression and other related ranking problems. There are three terms to be clarified: *ranking*, *sorting* and *multipartite ranking*.

Ranking generally refers to those problems where the algorithm is given a set of ordered labels [36], with one label for each pattern, and the objective is to learn a rule able to rank patterns by using this discrete set of labels. The induced ordering should be partial with respect to the patterns, in the sense that ties are allowed. This rule should be able to obtain a good ranking, but not to classify patterns in the correct class. For example, if the labels predicted by a classifier are shifted one category (in the ordinal scale) with respect to the actual ones, the classifier will still be a perfect ranker.

Another term, *sorting* [36] is referred to the problem where the algorithm is given a total order for the training dataset and the objective is to rank new sets during the test phase. As we can see, this is equivalent to a ranking problem where the size of the label set is equal to the number of training points, $Q = N$. Ties are not allowed for the prediction. Again, the interest is in learning a function that can give a total ordering of the patterns instead of a concrete label.

The *multipartite ranking* problem is a generalisation of the well-known bipartite ranking one. Multipartite ranking can be seen as an intermediate point between ranking and sorting. It is similar to ranking because training patterns are labelled with one of Q ordered ratings ($Q = 2$ for bipartite ranking), but here the goal is to learn from them a ranking function able to induce a total order

in accordance with the given training ratings [37]–[39], which is similar to sorting. The objective of multipartite ranking is to obtain a ranking function which ranks "high" classes ahead of "low" classes (in the ordinal scale), being this a refinement of the order information provided by an ordinal classifier, as the latter does not distinguish between objects within the same category. ROC analysis, which evaluates the ability of classifiers to sort positive and negative instances in terms of the area under the ROC curve, is a clear example of training a binary classifier to perform well in a bipartite ranking problem. The relationship between multipartite ranking and ordinal classification is discussed in [38]. An ordinal regression classifier can be used as a ranking function by interpreting the class labels as scores. However, this type of scoring will produce a large number of ties (which is not desirable for multipartite ranking). On the other hand, a multipartite ranking function $f(\cdot)$ can be turned into an ordinal classifier by deriving thresholds to define an interval for each class, but how to find the optimal thresholds is an open issue.

A more general term is *learning to rank*, gathering different methods in which the goal is to automatically construct a ranking model from training data [40]. Methods used for the three previously mentioned problems can be used for *learning to rank* ones. Moreover, ordinal regression can be used as a *learning to rank* algorithm, where the categories are individually evaluated for each training pattern, using a finite ordinal scale. In this context, we refer now to the categorisation presented in [40], which establishes different families of ranking model structures: *pointwise* or *itemwise ranking* (where the relevance of an input vector $\mathbf{x}$ is predicted by using either real-valued scores or ordinal labels), *pairwise ranking* (where the relative order between two input vectors $\mathbf{x}$ and $\mathbf{x}'$ is tried to be predicted, i.e. the local comparison nature of ranking, which can be easily cast to binary classification) and *listwise ranking* (where the algorithms try to order a finite set of patterns $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ by minimising the inconsistency between the predicted permutation and the training permutation). Ordinal regression methods are *pointwise ranking* models, where each vector is assigned an ordinal label in order to rank it. In this way, they can be used for ranking as an alternative to both *pairwise* and *listwise* structures, which have serious problems of scalability with the size of the training dataset [41], the former needing to examine all pairs of patterns and the latter considering all possible permutations of the training data.

In summary, ordinal regression is a pointwise approach to classify data, where the labels exhibit a natural order. It is related to the problems of ranking, sorting and multipartite ranking, but, during the test phase, its objective is to obtain correct labels or labels as close as possible to the correct ones, not a correct relative partial order of the patterns (ranking), a total order of patterns in accordance to the order of the training set (sorting) or a total order in accordance to the training labels (multipartite ranking).

2.3 Advanced related topics

In this section, other advanced methods related to ordinal regression are surveyed. They are outside the scope of this paper, as they consider different learning settings

Monotonic classification [42]–[44] is a special class of ordinal classification task, where there are monotonicity constraints between features and decision classes, i.e. $\mathbf{x} \succeq \mathbf{x}' \rightarrow f(\mathbf{x}) \geq f(\mathbf{x}')$ [45], where $\mathbf{x} \succeq \mathbf{x}'$ means that $\mathbf{x}$ dominates $\mathbf{x}'$, i.e. $x_k \geq x'_k, k = 1, \ldots, K$. Monotonic classification tasks are very common in real-world problems [43] (e.g. consider the case where employers must select their employees based on their education and experience), where monotonicity may be an important model requirement for justifying the decision made. This kind of problems have been approached, for example, by decision trees [43], [46] and rough set theory [44].

A recent work is concerned with transductive ordinal regression [27], where a SVM model is derived to learn from a set of labelled and unlabelled patterns. The core of their formulation is an objective function that caters to several commonly used loss functions in transductive settings, but for ordinal regression. This SVM model is combined with a proposed label swapping scheme for multiple class transduction to derive ordinal decision boundaries that pass through a low-density region of the augmented labelled and unlabelled data. Another related work [47] considers transfer learning in the same context, where the objective is to obtain a classifier for new target domains using the available label information of other related source domains. The proposed method spans the feasible solution space with an ensemble of ordinal classifiers from the multiple relevant source domains, using the maximum margin criterion.

Uncertainty has been included in ordinal regression models in two different ways. Nondeterministic ordinal classifiers (defined as those allowed to predict more than one label for some patterns) are considered in [48]. In [49] a kernel model is proposed for those ordinal problems where partial class memberships probabilities are available instead of crisp labels.

One step forward [50] considers those problems where the prediction labels follow a circular order (e.g. directional predictions).

3 An Ordinal Regression Taxonomy

In this section, a taxonomy of ordinal regression methods is proposed. With this purpose we firstly review what have been referred to as *naïve approaches*, in the sense that the model is obtained by using other standard machine learning prediction algorithms (e.g. nominal classification or standard regression). Secondly, *ordinal binary decomposition approaches* are reviewed, the main idea being to decompose the ordinal problem into several binary ones, which are separately solved by multiple models or by one multiple-output model. The third group will

include the set of methods known as *threshold models*, which are based on the general idea of approximating a real value predictor and then dividing the real line into intervals. The taxonomy proposed is given in Fig. 1.

3.1 Naïve approaches

Ordinal regression problems can be easily simplified into other standard problems, which generally involves making some assumptions. As will be later discussed, these methods can be very competitive given that, even though these assumptions may not hold, they inherit the performance of very well-tuned models.

3.1.1 Regression

One idea is to cast all the different labels $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$ into real values $\{r_1, r_2, \ldots, r_Q\}$ [51], where $r_i \in \mathbb{R}$, and then to apply standard regression techniques [2], [52], [53] (such as neural networks, support vector regression...). Typically, the value of each label is related to its position in the ordinal scale, i.e. $r_i = i$. For example, Kramer et al. [54] map the ordinal scale by assigning numerical values, applying a regression tree model and rounding the results for assigning the class when predicting new values. They also evaluate the possibility of using the median, the mode, or the rounded mean of all the patterns in the leaves of the tree. The main problem with these approaches is that real values used for the labels may hinder the performance of the regression algorithms, and there is no principled way of deciding the value a label should have without prior information about the problem, since the distance between classes is unknown. Moreover, regression learners will be more sensitive to the representation of the label rather than its ordering [55]. A recent alternative is proposed in [56], where, instead of choosing arbitrary ordered values for the different labels, the variable is reconstructed by examining the different pairwise class distances.

3.1.2 Nominal classification

Ordinal classification problems are usually considered from a standard nominal perspective, and the order between classes is simply ignored. Some researchers routinely apply nominal response data analysis methods (yielding results invariant to the permutation of the categories) to both nominal and ordinal target variables alike because they are both categorical [57]. Nominal classification algorithms ignore the ordering of the labels, thus requiring more training data [55]. The Support Vector Machine paradigm (SVM) [58] is perhaps the most common kernel learning method for statistical pattern recognition. Beyond the application of the kernel trick to allow non-linear decision discriminants, and the slack-variables to avoid inseparability, relax the constraints and handle noisy data, the original binary SVM had to be reformulated to deal with multiclass problems [59].

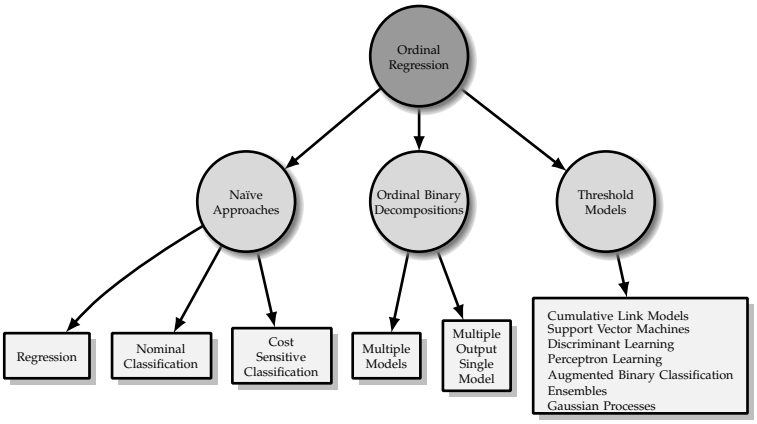


Fig. 1. Proposed taxonomy for ordinal regression methods

TABLE 1
Example of different cost matrices for a five class classification problems, with class labels $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}$.

Zero-one	Absolute cost	Quadratic cost
$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 4 & 9 & 16 \\ 1 & 0 & 1 & 4 & 9 \\ 4 & 1 & 0 & 1 & 4 \\ 9 & 4 & 1 & 0 & 1 \\ 16 & 9 & 4 & 1 & 0 \end{pmatrix}$

Actual class labels are arranged in rows, while predicted class labels are arranged in columns.

3.1.3 Cost-sensitive classification

A more advanced method that can be considered in this group is cost-sensitive learning. Many real-world applications of machine learning and data mining require the evaluation of the learned system with different costs for different types of misclassification errors [60]. This is the case with ordinal regression, although the exact costs for misclassification can not be always evaluated a priori. The cost of misclassifications can be forced to be different depending on the distance between real and predicted classes, in the ordinal scale. The work of Kotsiantis and Pintelas [61] considers cost-sensitive classification, by using absolute costs (i.e. the element c_{ij} of the cost matrix $\mathbf{C}$ is equal to the difference in the number of categories, $c_{ij} = |i - j|$). Different algorithms are shown to obtain better MAE values when cost matrices are used, without harming (in fact even improving) accuracy [61]. We include two cost matrices for a five class problem in Table 1, with the absolute cost matrix and the quadratic cost ($c_{ij} = |i - j|^2$), together with a zero-one cost matrix, which is the one assumed in nominal classification. Other possibilities are to choose asymmetric costs or non-convex two-Gaussian cost [41]. Again, the main problem is that, without a priori knowledge of the ordinal regression problem, it is not clear which cost matrix is more suitable.

TABLE 2
Binary decompositions for a 5-class ordinal problem, with class labels $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}$.

Nominal decompositions	
OneVsAll	*OneVsOne*
$\begin{pmatrix} +,-,-,-,- \\ -,+,-,-,- \\ -,-,+,-,- \\ -,-,-,+,- \\ -,-,-,-,+ \end{pmatrix}$	$\begin{pmatrix} -,-,-,-, , , , , , \\ +, , , ,-,-,-, , , \\ ,+, , ,+, , ,-,-, \\ , ,+, , ,+, ,+, ,- \\ , , , ,+, , ,+, ,+,+ \end{pmatrix}$

Ordinal decompositions			
OrderedPartitions	*OneVsNext*	*OneVsFollowers*	*OneVsPrevious*
$\begin{pmatrix} -,-,-,- \\ +,-,-,- \\ +,+,-,- \\ +,+,+,- \\ +,+,+,+ \end{pmatrix}$	$\begin{pmatrix} -, , , \\ +,-, , \\ ,+,-, \\ , ,+,- \\ , , ,+ \end{pmatrix}$	$\begin{pmatrix} -, , , \\ +,-, , \\ +,+,- \\ +,+,+,- \\ +,+,+,+ \end{pmatrix}$	$\begin{pmatrix} +,+,+,+ \\ +,+,+,- \\ +,+,-, \\ +,-, , \\ -, , , \end{pmatrix}$

3.2 Ordinal binary decompositions

This group includes all those methods which are based on decomposing the ordinal target variable into several binary ones, which are then estimated by a single or multiple models. A summary of the decompositions is given in Table 2, where five classes are considered, each method generating a different decomposition matrix. Columns of the matrix correspond to the binary subproblems and rows to the role of each class for each subproblem. The symbol $+$ is associated to the positive class and the symbol $-$ to the negative one. If the class is not used in the specific binary subproblem, no symbol is included in the corresponding position. *OneVsAll* and *OveVsOne* formulations are nominal classification methods (and should be listed as naïve approaches), but they have been included in this table for comparison purposes. Note the high number of binary decompositions needed by *OneVsOne* (in this case, 10 combinations).

Two main issues have to be taken into account when analysing the methods herein presented: 1) some of them are based on the idea of training a different model for each subproblem (multiple model approaches), while others learn one single model for all the subproblems; 2) apart from defining how to decompose the problem, it is important to define a rule for predicting new patterns, once the decision values are obtained. For the prediction phase, the corresponding binary codes of Table 2 can be considered as part of the error-correcting output codes (ECOC) framework [62], where the predicted class is the one closest to the code formed by all binary responses. Taking the first criterion into account, we have divided ordinal binary decomposition algorithms into *multiple model* and *multiple-output single model* approaches.

3.2.1 Multiple model approaches

Ordinal information gives us the possibility of comparing the different labels. For a given rank q, a direct question can be the following, "is the label of pattern $\mathbf{x}$ greater than q?" [41]. This question is clearly a binary classification problem, so ordinal classification can be solved by considering each binary classification problem independently and combining the binary outputs into a label, which is the approach followed by Frank and Hall in [63] (this decomposition is called *OrderedPartitions* in Table 2). In their work, Frank and Hall considered C4.5 as the binary classifier and the decision of the different binary classifiers were combined by using associated probabilities $p_q = P(y \succ \mathcal{C}_q|\mathbf{x}), q = 1, \ldots, Q-1$:

$$P(y = \mathcal{C}_1|\mathbf{x}) \approx 1 - p_1, P(y = \mathcal{C}_Q|\mathbf{x}) \approx p_{Q-1},$$
$$P(y = \mathcal{C}_q|\mathbf{x}) \approx p_{q-1} - p_q,\ 2 \leq q \leq Q-1.$$

Note that this approach may lead to negative probability estimates [64], given that binary classifiers are independently learned and nothing assures that $p_{q-1} < p_q$. When there is no need for proper probability estimations, prediction can be done by selecting the maximum.

In the work of Waegeman et al. [65], this framework is used but explicit weights over the patterns of each binary system are imposed, in such a way that errors on training objects are penalised proportionally to the absolute difference between their rank and q (the category examined). Additionally, labels for the test set are obtained by combining the estimated outcomes y_q of all the $Q-1$ binary classifiers. The interpretation of these binary outcomes $y_{qi} \in \{+1, -1\}, q = 1, \ldots, Q-1, i = 1, \ldots, N$, intuitively leads to $y_i \succ \mathcal{C}_q$ if $y_{qi} = +1$. In this way, the rank k is assigned to pattern $\mathbf{x}_i$ so that $y_{qi} = -1, \forall q < k$, and $y_{qi} = +1, \forall q \geq k$. As stated by the authors, this strategy can result in ambiguities for some test patterns, and they should be solved by using similar techniques to those considered for nominal classification. A very similar scheme is proposed in [12], where the weights are obtained slightly differently, and different kernels are used for the different binary classification sub-problems.

Other ordinal binary decompositions can be found in the literature. The cascade linear utility model [66] considers $Q-1$ projections, in such a way that projection q separates classes $\mathcal{C}_1 \cup \ldots \cup \mathcal{C}_{Q-q-1}$ from class C_{Q-q}, i.e. one class is eliminated for each projection (this is the *OneVsPrevious* decomposition in Table 2). The predictions are then combined by a union utility function. Finally, binary SVMs were also applied to ordinal regression [15], by making use of the ordinal pairwise partitioning approach [14]. This approach is composed of four different reformulations of the classical *OneVsOne* and *OneVsAll* paradigms. *OneVsNext* considers that each binary classifier q separates class $\mathcal{C}_q$ from class $\mathcal{C}_{q+1}$, and *OneVsFollowers* (which is similar to the *OneVsPrevious* approach in [66] but in the opposite direction) constructs each binary classifier q for the task of separating class $\mathcal{C}_q$ from classes $\mathcal{C}_{q+1} \cup \ldots \cup \mathcal{C}_Q$. The prediction phase is then approached by examining each binary classifier in order, so that, if a model predicts that the pattern is in the class which is isolated (not grouped with other classes), then this is the predicted class. This can be done in a forward manner or in a backward manner [15].

Finally, another possibility [67] is to derive a classifier for each class but separating the labels into groups

of three classes (instead of only two) for intermediate subtasks (labels lower than $\mathcal{C}_q$, label $\mathcal{C}_q$, and labels higher than $\mathcal{C}_q$), or two classes for the extreme ones. The objective is to incorporate the order information in the subclassification tasks. Although the decomposition for intermediate classes is not binary but ternary, this approach has been included in this group because its motivation is similar to all the aforementioned.

3.2.2 Multiple-output single model approaches

Among non-parametric models, one appealing property of neural networks is that they can handle multiple responses in a seamless fashion [68]. Usually, as many output neurons as the number of target variables are included in the output layer and targets are presented to the network in the form of vectors $\mathbf{t}_i, i = 1, \ldots, N$. When applied to nominal classification, the most usual approach is to consider a 1-of-Q coding scheme [53], i.e. $\mathbf{t}_i = \{t_{i1}, \ldots, t_{iQ}\}$, $t_{iq} = 1$ if $\mathbf{x}_i$ corresponds to an example belonging to class $\mathcal{C}_q$, and $t_{iq} = 0$ (or $t_{iq} = -1$), otherwise. In the ordinal regression framework, one can take the ordering information into account to design specific ordinal target coding schemes, which can improve the performance of the methods. Indeed, all the decompositions in Table 2 can be used to train neural networks, by taking each row as the code for the target class, $\mathbf{t}_i$, and a single model will be obtained for all related subproblems (considering that each output neuron is solving each subproblem). This can be done by assigning a value ($+1$, 0 or -1) to each of the different symbols ($+$ or $-$) in Table 2. For sigmoidal output neurons, a 1 is assigned for positive symbols ($+$) and a 0 for negative ones ($-$). For hyperbolic functions, negative symbols are represented with a -1 and positive ones also with a 1. Those decompositions where a class is not involved should be treated as a "does not matter" condition where, whatever the output response, no error signal should be generated [69].

A generalisation of ordinal perceptron learning [70] in neural networks was proposed in [71]. The method is based on two main ideas: 1) the targets are coded using the *OrderedPartitions* approach; and 2) instead of using the softmax function [53] for the output nodes, a standard sigmoid function is imposed, and the category assigned to a pattern is equal to the index previous to that of the first output node whose value is higher than a predefined threshold T, or when no nodes are left. This method ignores inconsistencies (e.g. a sigmoid with value higher than T after the index selected).

Extreme learning machines (ELMs) are single-layer feedforward neural networks, where the hidden layer does not need to be tuned given that corresponding weights are randomly assigned. ELMs have demonstrated good scalability and generalisation performance with a faster learning speed when compared to other models such as SVMs [72]. They have been adapted to ordinal regression [73], and one of the proposed ordinal ELMs also considers *OrderedPartitions* targets. Additionally, multiple models are also trained using the *OneVsOne* and the *OrderedPartitions* approaches. For the prediction phase, the loss-based decoding approach [62] is utilised, i.e. the chosen label is that which minimises the exponential loss, $k = \arg\min_{q=1,\ldots,Q} d\left(\mathbf{M}_q, \mathbf{y}(\mathbf{x})\right)$, where $\mathbf{M}_q$ is the code associated to class q (q-th row of the coding matrix), $\mathbf{y}(\mathbf{x})$ is the vector of predictions, and $d\left(\mathbf{M}_q, \mathbf{y}(\mathbf{x})\right)$ is the exponential loss function, $d\left(\mathbf{M}_q, \mathbf{y}(\mathbf{x})\right) = \sum_{i=1}^{Q} \exp\left(\mathbf{M}_{qi} \cdot \mathbf{y}_i(\mathbf{x})\right)$. The values of the vector $\mathbf{y}(\mathbf{x})$ are assumed to be in the $[-1, +1]$ range, and those of $\mathbf{M}_q$ in the set $\{-1, 0, +1\}$. The single ELM was found to obtain slightly better generalisation results and also to report the lowest computational time [73]. Other adaption of the ELM is found in [74], where an evolutionary algorithm is applied to optimise the different weights of the model by using a fitness function to impose the ordering restriction in model selection. A different approach is taken in [75], where the ordinal constraints are included into the weights connecting the hidden and output layers.

Costa [69] followed a probabilistic framework to propose another neural network architecture able to exploit the ordinal nature of the data. The proposal is based on the joint prediction of constrained concurrent events, which can be turned into a classification task defined in a suitable space through a "partitive approach". An appropriate entropic loss is derived for $\mathbf{P}(\mathcal{Y})$, i.e. the set of subsets of $\mathcal{Y}$, where $\mathcal{Y}$ is a set of Q elementary events. A probability for each possible subset should be estimated, leading to a total of 2^Q probabilities. However, depending on the classification problem, not all possibilities should be examined. For example, this is simplified for random variables taking values in finite ordered sets (i.e. ordinal regression), as well as in the case of independent boolean random variables (i.e. nominal classification). To adapt neural networks to the ordinal case structure, targets were reformulated following the *OneVsFollowers* approach and the prediction phase was accomplished by considering that, under its constrained entropic loss formulation, the output of the q-th output neuron estimates the probability that q and $q-1$ events are both true. This methodology was further evaluated and compared in other works [64], [76], [77].

Although all these neural network approaches consist of a single model, they are trained independently in the sense that the output of the neurons do not depend on the other outputs (only on common nonlinear transformations of the inputs). That is the reason why we have included them into the category of ordinal binary decompositions.

These neural network models can be grouped under the term multitask learning [78] (MTL), which is a learning paradigm that considers the case of simultaneously tackling several related tasks. Any of the different proposals in this field could be applied to train a single model for the different ordinal decompositions analysed in this section. Indeed, one of the existing proposals, MTL via conic programming [79], was validated in the

context of ordinal regression, showing promising results.

3.3 Threshold models

Often, in the ordinal regression paradigm, it is natural to assume that an unobserved continuous variable underlies the ordinal response variable. Such a variable is called a latent variable, and methods based on that assumption are known as threshold models, which are the most popular approaches for modelling ordinal and ranking problems [49]. These methodologies estimate:

- A function $f(\mathbf{x})$ that tries to predict the values of the latent variable, acting as a mapping function from feature space to the real one (similar to the ranking function to be learned by multipartite algorithms).
- A set of thresholds $\mathbf{b} = (b_1, b_2, \dots, b_{Q-1}) \in \mathbb{R}^{Q-1}$ to represent intervals in the range of $f(\mathbf{x})$, which must satisfy the constraints $b_1 \leq b_2 \leq \dots \leq b_{Q-1}$.

Threshold models can be seen as an extension of naïve regression models. The main difference between these two approaches is that the distances among the different classes are not defined a priori for threshold models, being estimated during the learning process. Although they are also related to (single-model) ordinal binary decomposition approaches, the main difference is that threshold models are based on one single projection vector with multiple thresholds, one for each class.

3.3.1 Cumulative link models

Arising from a statistical background, the Proportional Odds Model (POM) is one of the first models specifically designed for ordinal regression [80], dated back to 1980. It is a member of a wider family of models recognised as Cumulative Link Models (CLMs) [81]. In order to extend binary logistic regression to ordinal regression, CLMs predict probabilities of groups of contiguous categories, taking the ordinal scale into account. In this way, cumulative probabilities $P(y \preceq \mathcal{C}_j|\mathbf{x})$ are estimated, which can be directly related to standard probabilities:

$$P(y \preceq \mathcal{C}_q|\mathbf{x}) = P(y = \mathcal{C}_1|\mathbf{x}) + \dots + P(y = \mathcal{C}_q|\mathbf{x}),$$
$$P(y = \mathcal{C}_q|\mathbf{x}) = P(y \preceq \mathcal{C}_q|\mathbf{x}) - P(y \preceq \mathcal{C}_{q-1}|\mathbf{x}),$$

with $q = 2, \dots, Q$, and considering by definition that $P(y = \mathcal{C}_1|\mathbf{x}) = P(y \preceq \mathcal{C}_1|\mathbf{x})$ and $P(y \preceq \mathcal{C}_Q|\mathbf{x}) = 1$.

A decision rule $f : \mathcal{X} \to \mathcal{Y}$ is not fitted directly. Instead, stochastic ordering of space $\mathcal{X}$ is satisfied by the following general model form [28]:

$$g^{-1}\left(P(y \preceq \mathcal{C}_q|\mathbf{x})\right) = b_q - \mathbf{w}^T\mathbf{x}, q = 1, \dots, Q,$$

where $g^{-1} : [0, 1] \to (-\infty, +\infty)$ is a monotonic function often referred to as the inverse link function and b_q is the threshold defined for class $\mathcal{C}_q$. This model is clearly inspired by the latent variable motivation, considering that $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ is a linear transformation. Consider the error of the model of the latent variable, $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \epsilon$, where ϵ is the random component with zero expectation, $\mathbf{E}[\epsilon] = 0$, distributed according to F_ϵ. If a distribution assumption F_ϵ is made for ϵ, the cumulative model is obtained by choosing the inverse distribution F_ϵ^{-1} as the inverse link function g^{-1}. The most common choice for the distribution of ϵ is the logistic function (which is indeed the one selected for the POM [82]), although probit, complementary log-log, negative log-log or cauchit functions could also be used [81]. If the ordinal response is a coarsely measured latent continuous variable $f(\mathbf{x})$, label $\mathcal{C}_q$ in the training set is observed if and only if $f(\mathbf{x}) \in [b_{q-1}, b_q]$, where the function f and $\mathbf{b} = (b_0, b_1, ..., b_{Q-1}, b_Q)$ are to be determined from the data. It is assumed that $b_0 = -\infty$ and $b_Q = +\infty$, so the real line, defined by $f(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, is divided into Q consecutive intervals. Each region separated by two consecutive biases corresponds to a category $\mathcal{C}_q$. The constraints $b_1 \leq b_2 \leq \dots \leq b_{Q-1}$ ensure that $P(y \preceq \mathcal{C}_q|\mathbf{x})$ increases with q [83].

As will be seen, all the models in this section are inspired by the POM in the strategy assumed, obtaining a one-dimensional mapping function and dividing the real line into different ordered intervals. This mapping function can be used to obtain more information about the confidence of the predictions by relating it to its proximity to the biases. Additionally, the POM model provides us with a solid probabilistic interpretation. The distribution of ϵ is assumed to be the standard logistic function for the POM:

$$g^{-1}\left(P(y \preceq \mathcal{C}_q|\mathbf{x})\right) = \ln\left(\frac{P(y \preceq \mathcal{C}_q|\mathbf{x})}{P(y \succ \mathcal{C}_q|\mathbf{x})}\right) = b_q - \mathbf{w}^T\mathbf{x},$$

where $q = 1, \dots, Q-1$, $odds(y \preceq \mathcal{C}_q|\mathbf{x}) = \exp(b_q - \mathbf{w}^T\mathbf{x})$, so $odds(y \preceq \mathcal{C}_q|\mathbf{x}) = \frac{P(y \preceq \mathcal{C}_q|\mathbf{x})}{1 - P(y \preceq \mathcal{C}_q|\mathbf{x})}$. Therefore, the ratio of the odds for two pattens $\mathbf{x}_0$ and $\mathbf{x}_1$ are proportional:

$$\frac{odds(y \preceq \mathcal{C}_q|\mathbf{x}_1)}{odds(y \preceq \mathcal{C}_q|\mathbf{x}_0)} = \exp(-\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_0)).$$

More flexible non-proportional alternatives have been developed, one of them simply assuming different $\mathbf{w}$ for each class (which is known as the generalised ordered logit model [84]). Another alternative applies the proportional odds assumption only to a subset of variables (partial proportional odds [85]). Moreover, Tutz [86] presented a general framework for parametric models that extends generalised additive models to incorporate nonparametric parts.

Apart from assuming proportional odds, linear CLMs are rather inflexible since the decision functions are always linear hyperplanes, this generally affecting the performance of the model (as analysed in the experimental section of this work). A non-linear version of the POM model was proposed in [18], [83] by simply setting the projection $f(\mathbf{x})$ to be the output of a neural network. The probabilistic interpretation of CLMs can be used to apply a maximum likelihood maximisation for setting the network parameters. Gradient descent techniques with proper constraints for the biases serve this purpose. This non-linear generalisation of the POM model based on neural networks was considered in [87], where an

evolutionary algorithm was applied to optimise all the parameters considered. Linear ordinal logistic regression was combined with nonlinear kernel machines using primal-dual relations from Nystrom sampling [88]. However, to make the computation of the model feasible, a sub-sample from the data had to be selected, which limits the applicability to those cases where there is a reasonable way to do this [88].

An interesting alternative to CLMs is the so-called ordistic model presented in [89]. The work presents two threshold-based constructions which can be used to generalise loss functions for binary labels, such as the logistic and hinge loss, and another generalisation of the logistic loss based on a probabilistic model for ordered labels. Both constructions are based on including $Q-1$ thresholds partitioning the real line to Q segments, but they differ in how predictors outside the "correct" segment (or too close to its edges) are penalised. The immediate-threshold construction only penalises the violations of the two thresholds limiting this segment, while the all-threshold one considers all of them.

3.3.2 Support vector machines

Because of their good generalisation performance, SVM models are maybe the most widely applied ones to ordinal regression, their structure being easily adapted to that of threshold models. The proposal of Herbrich et al. [28], [90] is the first SVM based algorithm, where they consider a pairwise approach by deriving a new dataset made up of all possible difference vectors $\mathbf{x}_{ij}^d = \mathbf{x}_i - \mathbf{x}_j$ and $y_{ij} = \text{sign}\left(\mathcal{O}(y_i) - \mathcal{O}(y_j)\right)$, with $y_i, y_j \in \{\mathcal{C}_1, \ldots, \mathcal{C}_Q\}$. In contrast, all the SVM pointwise approaches share the common objective of seeking $Q-1$ parallel discriminant hyperplanes, all of them represented by a common vector $\mathbf{w}$ and the scalars biases $b_1 \leq \ldots \leq b_{Q-1}$ to properly separate training data into ordered classes. In this sense, several methodologies for the computation of $\mathbf{w}$ and $\{b_1, \ldots, b_{Q-1}\}$ can be considered. The work of Shashua and Levin [91] introduced two first methods: the maximisation of the margin between the closest neighbouring classes and the maximisation of the sum of margins between classes. Both approaches present two main problems [64]: the model is incompletely specified, because the thresholds are not uniquely defined, and they may not be properly ordered at the optimal solution, since the inequality $b_1 \leq b_2 \leq \ldots \leq b_{Q-1}$ is not included in the formulation.

Consequently, Chu and Keerthi [29], [92] proposed two different reformulations for the same idea, solving the problem of unordered thresholds at the solution. On the one hand, they imposed explicit constraints on the optimisation problem, only considering adjacent labels for threshold determination (Support Vector Ordinal Regression with Explicit Constraints, SVOREX). On the other hand, patterns in all the categories were allowed to contribute errors for each hyperplane (SVOR with Implicit Constraints, SVORIM), which, as they prove [29], leads to automatically satisfied constraints in the optimal solution (see Lemma 1 of [29]). Let N_q be the number of patterns of class $\mathcal{C}_q$, and let $\mathbf{x}_i^q$ be those patterns $\mathbf{x}$ which class label is $\mathcal{C}_q$. The SVORIM learning problem is defined as follows:

$$\min_{\mathbf{w},\mathbf{b},\xi,\xi^*} \frac{1}{2}||\mathbf{w}|| + C\sum_{q=1}^{Q-1}\left(\sum_{j=1}^{q}\sum_{i=1}^{N_q}\xi_{ji}^q + \sum_{j=q+1}^{Q}\sum_{i=1}^{N_q}\xi_{ji}^{*q}\right),$$

subject to the constraints:

$$\begin{aligned} &\mathbf{w}\cdot\mathbf{x}_i^j - b_q \leq -1 + \xi_{ji}^q,\ \xi_{ji}^q \geq 0, && j \in \{1,\ldots,q\},\\ &\mathbf{w}\cdot\mathbf{x}_i^j - b_q \geq +1 - \xi_{ji}^{*q},\ \xi_{ji}^{*q} \geq 0, && j \in \{q+1,\ldots,Q\}, \end{aligned}$$

where $i \in \{1,\ldots,N_q\}$, $\mathbf{b} \in \mathbb{R}^{Q-1}$, ξ_{ji}^q and ξ_{ji}^{*q} are the slacks for the q-th parallel hyperplane (defined for the left and right part of the hyperplanes, respectively). The first group of constraints is focused on the left part of the j-th hyperplane (classes with $q \leq j$), while the second one is focused on the right part (classes with $q > j$). They empirically found that SVOREX performed better in terms of accuracy (with a more local behaviour), and SVORIM preceded in terms of absolute deviations in number of classes or MAE (with a more global behaviour), and this is justified theoretically based on the loss minimised for each method. The framework of reduction [41] also explains this from the point of view of the cost matrices selected. Our results seem to agree with these conclusions for discretised regression datasets, but the differences are not so clear for real ordinal regression ones. Generalisation properties for some ordinal regression algorithms, including SVOR, were further studied in [93].

In [94], the errors of an ordinal SVM classifier are studied separately depending on whether they correspond to upgrading errors (predicted label higher than the actual one) or downgrading ones (the predicted label being lower than the actual one). Authors address the two-objective problem of finding a classifier maximising simultaneously the two margins, and they show that the whole set of Pareto-optimal solutions can be obtained by solving a quadratic optimisation problem.

Some recent works focused on solving the bottleneck of these SVM proposals, which is usually the high computational complexity to handle larger datasets. Concerning this topic, two different proposals can be distinguished: block-quantised support vector ordinal regression [95] and ordinal-class core vector machines [96]. The former is based on performing kernel k-means and applying SVOR in the cluster representatives, on the idea of approximating the kernel matrix K by $\tilde{K}$ which will be composed of k^2 constant blocks, in such a way that the problem scales with the number of clusters, instead of the dataset size. The latter is an extension of core vector machines [97] in the ordinal regression setting. Finally, an incremental version of SVOR algorithms is proposed in [98].

3.3.3 Discriminant learning

Discriminant learning has also been reformulated to tackle ordinal regression [31]. Discriminant analysis is usually not considered as a classification technique by itself, but rather as a supervised dimensionality reduction. Nonetheless, it is widely used for that purpose, since, as a projection method, the definition of thresholds can be used to discriminate the classes. In general, to allow the computation of the optimal projection for the data, this algorithm analyses two main objectives: the maximisation of the between-class distance, and the minimisation of the within-class distance, by using variance-covariance matrices and the Rayleigh coefficient. In order to reformulate the algorithm for ordinal regression, an ordering constraint over contiguous classes is imposed on the averages of projected patterns of each class, which leads the algorithm to order projected patterns according to their label. This will preserve the ordinal information and avoid some serious ordinal misclassification errors. The original optimisation problem is transformed and extended with a penalty term (C):

$$\min J(\mathbf{w}, \rho) = \mathbf{w}^{\mathrm{T}} S_w \mathbf{w} - C\rho,$$

subject to $\mathbf{w}^{\mathrm{T}}(\mathbf{m}_{q+1} - \mathbf{m}_q) \geq \rho$, where $\mathbf{m}_q = \frac{1}{N_q}\sum_{i=1}^{N_q} \mathbf{x}_i$, S_w is the within-class scatter matrix and ρ represents the minimum difference of the projected means between consecutive classes (if $\rho > 0$, the projected means are correctly ranked). This methodology is known as Kernel Discriminant Learning for Ordinal Regression (KDLOR) [31] and it has been used in some later works [9], [99]. In [100], the KDLOR model is extended by trying to learn multiple orthogonal projections, which are then combined into a final decision function.

The method was extended in [101], [102] based on the idea of preserving the intrinsic geometry of the data in the embedded non-linear structure, i.e. in the induced high-dimensional feature space, via kernel mapping. This consideration is the basis of manifold learning [53], and the algorithms mentioned construct a neighbourhood graph (which takes the ordinal nature of the dataset into account) which is afterwards used to derive the Laplacian matrix and obtain a projection which considers the underlying manifold of the data. A related method is proposed in [103], where several different projections are iteratively derived.

3.3.4 Perceptron learning

PRank [104] is a perceptron online learning algorithm with the structure of threshold models. It was then extended by approximating the Bayes point, what provides good performance for generalisation [55].

3.3.5 Augmented binary classification

Although the approaches in Subsection 3.2 are simple to implement, their generalisation performance cannot be analysed easily. The two algorithms included in this subsection work differently, and, as later analysed, the models derived are equivalent to threshold models.

TABLE 3
Extended binary transformation for three given patterns $(\mathbf{x}_1, y_1 = \mathcal{C}_1), (\mathbf{x}_2, y_2 = \mathcal{C}_2), (\mathbf{x}_3, y_3 = \mathcal{C}_3)$, the identity coding matrix and the quadratic cost matrix.

			$\mathbf{x}_i^{(q)}$		
i	q	$w_{i,q}$	$\mathbf{x}$	$\mathbf{m}_q$	$y_i^{(q)}$
1	1	$2 \cdot \lvert 0-1 \rvert = 2$	$\mathbf{x}_1$	$\{1,0\}$	$2[\![1<1]\!]-1=-1$
1	2	$2 \cdot \lvert 1-4 \rvert = 6$	$\mathbf{x}_1$	$\{0,1\}$	$2[\![2<1]\!]-1=-1$
2	1	$2 \cdot \lvert 1-0 \rvert = 2$	$\mathbf{x}_2$	$\{1,0\}$	$2[\![1<2]\!]-1=+1$
2	2	$2 \cdot \lvert 0-1 \rvert = 2$	$\mathbf{x}_2$	$\{0,1\}$	$2[\![2<2]\!]-1=-1$
3	1	$2 \cdot \lvert 4-1 \rvert = 6$	$\mathbf{x}_3$	$\{1,0\}$	$2[\![1<3]\!]-1=+1$
3	2	$2 \cdot \lvert 1-0 \rvert = 2$	$\mathbf{x}_3$	$\{0,1\}$	$2[\![2<3]\!]-1=+1$

A reduction framework can be found in the works of Lin and Li [41], [105], where ordinal regression is reduced to binary classification by applying three steps:

1) A coding matrix $\mathbf{M}$ is used to represent the class being examined. Given a coding matrix $\mathbf{M}$ of $(Q-1)$ rows, input patterns $(\mathbf{x}_i, y_i)$ are transformed into extended binary patterns by replicating them, $(\mathbf{x}_i^{(q)}, y_i^{(q)})$, with:

$$\mathbf{x}_i^{(q)} = (\mathbf{x}_i, \mathbf{m}_q),\ y_i^{(q)} = 2[\![q < \mathcal{O}(y_i)]\!] - 1,$$

where $q = 1, \ldots, Q-1$, $\mathbf{m}_q$ is the q-th row of $\mathbf{M}$ and $[\![\cdot]\!]$ is a Boolean test which is 1 if the inner condition is true, and 0 otherwise. $Q-1$ replicates of each pattern are generated with weights:

$$w_{i,q} = (Q-1) \cdot |C_{\mathcal{O}(y_i),q} - C_{\mathcal{O}(y_i),q+1}|,$$

where $i = 1, \ldots, N$, C is a V-shaped cost matrix (i.e. $C_{\mathcal{O}(y_i),q-1} \geq C_{\mathcal{O}(y_i),q}$, if $q \leq \mathcal{O}(y_i)$, and $C_{\mathcal{O}(y_i),q} \leq C_{\mathcal{O}(y_i),q+1}$, if $q \geq \mathcal{O}(y_i)$). The cost matrix must be defined a priori. An example of this transformation is given in Table 3. As can be seen, the final extended pattern represents the question "Is the rank of $\mathbf{x}$ greater than q?" [41]. The weights measure the importance of the pattern for the binary classifier, and they are also used for the theoretical analysis.

2) A single binary classifier with confidence outputs, $f(\mathbf{x}, \mathbf{m}_k)$, is trained for the new weighted extended patterns, aiming at a low weighted $0/1$ loss.

3) A classification rule like the following is used to construct a final prediction for new patterns:

$$r(\mathbf{x}) = 1 + \sum_{q=1}^{Q-1} [\![f(\mathbf{x}, \mathbf{m}_q) > 0]\!]. \quad (1)$$

All the binary classification problems are solved jointly by computing a single binary classifier. The most striking characteristic of this algorithm is that it unifies many existing ordinal regression algorithms [41], such as the perceptron ones [104], kernel ranking [36], AdaBoost.OR [106], ORBoost-LR and ORBoost-All thresholded ensemble models [107], CLM [81] or several ordinal SVM

proposals (oSVM [64], SVORIM and SVOREX [29]). Moreover, it is important to highlight the theoretical guarantees provided by the framework, including the derived cost and regret bounds and the proof of equivalence between ordinal regression and binary classification. An extension of this reduction framework was proposed in [108], where ordinal regression is proved to be equivalent to a regular multiclass classification whose distribution is changed. This extension is free of the following restrictions: target functions should be rank-monotonic; and rows of loss matrix are convex.

The data replication method of Cardoso et al. [64] (whose previous linear version appeared in [10]) is a very similar framework, except that it essentially considers the absolute cost, consequently being less flexible. However, for ordinal regression, increasing the error with the absolute difference between the predicted and estimated labels is a natural choice in the absence of any other information [18]. An advantage of the framework of data replication is that it includes a parameter s which limits the number of adjacent classes considered, in such a way that the replicate q is constructed by using the $q-s$ classes to its 'left' and the $q+s$ classes to its 'right' [64]. This parameter $s \in \{1, ..., Q-1\}$ plays the role of controlling the increase of data points.

It is worth mentioning that augmented binary classification models and threshold models are closely related, and that is the reason why they have been included in this section. The extended patterns only differ in the new variables introduced by the coding matrix $\mathbf{M}$. The original version of the dataset is replicated in different subspaces, with different values for the new variables. By obtaining the intersection of the binary hyperplane in the extended dataset with each of the subspace replicas we derive parallel boundaries in the original dataset [64], with a single projection vector and multiple thresholds. In fact, SVORIM and reduction to SVM is known to be not so different in formulation [103]. The model of Mathieson [18] (threshold model) is equivalent to the one proposed in [64] (oNN, an augmented binary classification model) if the activation function of the output node is set to the *logsig* function and the model is trained to predict the posterior probabilities when fed with the original input variables and the variables generated by the data replication method. The predicted thresholds would be the weights of the connection of the added $Q-2$ components. Finally, augmented binary classification and ordinal binary decomposition are not disjoint categories. The former class of models do not restrict consistency of binary classifiers, making use of a "voting" of the binary classifiers (see Eq. 1 of [105]). Moreover, all the ordinal decompositions in Table 2 can be viewed as a special case of "cost-sensitive ordinal classification" via augmented binary classification [41].

3.3.6 Ensembles

From a different perspective, the confidence of a binary classifier can be regarded as an ordering preference. RankBoost [109] is a boosting algorithm that constructs an ensemble of those confidence functions to form a better ordering preference. Some efforts were made to apply a similar idea for ordinal regression problems, deriving into Ordinal Regression Boosting (ORBoost) [107]. The corresponding thresholded-ensemble models inherit the good properties of ensembles, including more stable predictions and sufficient power for approximating complicated target functions [110]. The model is composed of confidence functions, and their weighted linear combination is used as the projection $f(\mathbf{x})$. A set of thresholds for this projection is also included in the model and iteratively updated with the rest of parameters. Following a similar approach to [89], large margin bounds of the classification error and the absolute error are derived, from which two algorithms are presented: ORBoost with all margins and ORBoost with left-right margins [107]. Two alternative thresholded-ensemble algorithms are presented in [111], both generating an ensemble of ordinal decision rules based on forward stagewise additive modelling.

With a different perspective, the well-known AdaBoost algorithm was recently extended to improve any base ordinal regression algorithm [106]. The extension, AdaBoost.OR, proved to inherit the good properties of AdaBoost, improving both the training and test performances of existing ordinal classifiers. Another ordinal regression version of AdaBoost is proposed in [112], while in this case the adaption is based on considering a cost matrix both in pattern weighting and error updating.

The framework of negative correlation learning (where the ensemble members are learnt in such a way that the correlation between their responses is minimised) was used in the context of ordinal regression [17], [113] by calculating the correlation between the latent variable estimations or, alternatively, between the probabilities obtained by the ensemble members.

3.3.7 Gaussian processes

All the previous threshold models can be considered discriminative models in the sense that they estimate directly the posterior $P(y|\mathbf{x})$, or learn a function to map the input $\mathbf{x}$ to class labels. On the contrary, generative models learn a model of the joint probability $P(\mathbf{x}, y)$ of input patterns $\mathbf{x}$ and label y, and make the prediction by a Bayesian framework to estimate $P(y|\mathbf{x})$.

Gaussian Processes for Ordinal Regression (GPOR) [30] models the latent variable $f(\mathbf{x})$ using Gaussian Processes, to estimate then all the parameters by means of a Bayesian framework. The values of the latent function $\{f(\mathbf{x}_i)\}$ are assumed to be the given by random variables indexed by their input vectors in a zero-mean Gaussian process. Mercer kernel functions approximate the covariance between the functions of two input vectors. Finally, the thresholds are included in the model to divide the latent variable in consecutive intervals, and they are optimised together with the rest of parameters, using padding variables to avoid unordered solutions. Given

the latent function f, the joint probability of observing the ordinal variables is $P(D|f) = \prod_{i=1}^{N} P(y_i|f(\mathbf{x}_i))$, and the Bayes theorem is applied to write the posterior probability $P(f|D) = \frac{1}{P(D)} \prod_{i=1}^{N} P(y_i|f(\mathbf{x}_i))P(f)$. A Gaussian noise with zero mean and unknown variance σ^2 is assumed for the latent functions. The normalisation factor $P(D)$, more exactly $P(D|\boldsymbol{\theta})$, is known as the evidence for the vector of hyperparameters $\boldsymbol{\theta}$ and is estimated in the paper by two different approaches: a Maximum a Posteriori approach with Laplace approximation and an Expectation Propagation with variational methods. A more general GPOR was then proposed to tackle multi-class classification problems but with a free structure of preferences over the labels [114]. A probabilistic sparse kernel model was proposed for ordinal regression in [115], where a Bayesian treatment was also employed to train the model. A prior over the weights governed by a set of hyperparameters was imposed, inspired by the well-known relevance vector machine. Srijith et al. have proposed a probabilistic least squares version of GPOR [116], two different sparse versions [117] and a semi-supervised version [118].

3.4 Other approaches and problem formulations

This subsection includes some methods that are difficult to consider in the previous groups. For example, an alternative methodology is proposed by da Costa et al. [76], [77] for training ordinal regression models. The main assumption of their proposal is that the random variable class associated with a given pattern should follow a unimodal distribution. For this purpose, they provide two possible implementations: a parametric one, where a specific discrete distribution is assumed and the associated free parameters are estimated by a neural network; and a non-parametric one, where no distribution is assumed but the error function is modified to avoid errors from distant classes. The same idea was then applied to SVMs in [119] by solving an ordinal problem through a single optimisation process (the all-at-once strategy).

In [120], both decision trees and nearest neighbour (NN) classifiers are applied to ordinal regression problems by introducing the notion of consistency: a small change in the input data should not lead to a 'big jump' in the output decision, i.e. adjacent decision regions should have equal or consecutive labels. This rationale was used as a post-processing mechanism of a standard decision tree and as a pre- or post- processing step for the NN method. An improvement was presented in [121] to reduce the over-regularised decision region artifact by using ensemble learning techniques.

Two ordinal learning vector quantisation schemes, with metric learning, specifically designed for classifying data items into ordered classes, are introduced in [122], [123]. The methods use the order information during training, both in the selection of the prototypes and for determining the way they are updated.

Different prediction methods as a function of the error measure to be minimised are presented in [124]. The paper discusses the fact that the Bayes optimal decision for a classifier which return probability estimates is different depending on the loss function considered for the errors. In this way, for the maximisation of the accuracy one should consider the mode (or maximum probability), but the median of the probability distribution is the optimal decision when minimising the MAE in ordinal regression problems.

4 Experimental Study

4.1 Experimental design

In this subsection, the experiments are clearly specified, including the datasets and algorithms considered, the parameters to optimise, the performance measures and the statistical tests used for assessing the differences.

4.1.1 Datasets selected

The most widely used dataset repository is the one provided by Chu et al. [30], including different regression benchmark datasets. These datasets are not real ordinal classification problems but regression ones, which are turned into ordinal classification by discretising the target into Q different bins with equal frequency. It is clear that these datasets do not exhibit some characteristics of typical complex classification tasks, such as class imbalance, given that all classes are assigned the same number of patterns. However, we find interesting to check how the algorithms perform in this more controlled environment and to compare the conclusions obtained.

Table 4 shows the characteristics of the 41 datasets, including the number of patterns, attributes and classes, and also the number of patterns per class. The real ordinal classification datasets were extracted from benchmark repositories[1] (UCI [125] and `mldata.org` [126]), and the regression ones were obtained from the website of W. Chu[2]. For the discretised datasets, we considered $Q = 5$ and $Q = 10$ bins to evaluate the response of the classifiers to the increase in the complexity of the problem. The synthetic `toy` dataset was generated as proposed in [77] with 300 patterns. All nominal attributes were transformed into as many binary attributes as the number of categories, and all the datasets were properly standardised.

4.1.2 Algorithms selected

We have selected some representatives of the different families included in the proposed taxonomy (see Table 5). It is important to note that naïve approaches and ordinal binary decompositions can be applied using almost any base binary classifier or regressor. In our experiments, we have selected in those cases SVMs, given

1. We would like to note that many of these datasets have been previously considered in machine learning literature, but ignoring the ordering information.

2. http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html

TABLE 4
Characteristics of the benchmark datasets

Discretised regression datasets				
Dataset	#Pat.	#Attr.	#Classes	Class distribution
pyrim5 (P5)	74	27	5	≈ 15 per class
machine5 (M5)	209	7	5	≈ 42 per class
housing5 (H5)	506	14	5	≈ 101 per class
stock5 (S5)	700	9	5	140 per class
abalone5 (A5)	4177	11	5	≈ 836 per class
bank5 (B5)	8192	8	5	≈ 1639 per class
bank5′ (BB5)	8192	32	5	≈ 1639 per class
computer5 (C5)	8192	12	5	≈ 1639 per class
computer5′ (CC5)	8192	21	5	≈ 1639 per class
cal.housing5 (CH5)	20640	8	5	4128 per class
census5 (CE5)	22784	8	5	≈ 4557 per class
census5′ (CEE5)	22784	16	5	≈ 4557 per class
pyrim10 (P10)	74	27	10	≈ 8 per class
machine10 (M10)	209	7	10	≈ 21 per class
housing10 (H10)	506	14	10	≈ 51 per class
stock10 (S10)	700	9	10	70 per class
abalone10 (A10)	4177	11	10	≈ 418 per class
bank10 (B10)	8192	8	10	≈ 820 per class
bank10′ (BB10)	8192	32	10	≈ 820 per class
computer10 (C10)	8192	12	10	≈ 820 per class
computer10′ (CC10)	8192	21	10	≈ 820 per class
cal.housing (CH10)	20640	8	10	2064 per class
census10 (CE10)	22784	8	10	≈ 2279 per class
census10′ (CEE10)	22784	16	10	≈ 2279 per class
Real ordinal regression datasets				
Dataset	#Pat.	#Attr.	#Classes	Class distribution
contact-lenses (CL)	24	6	3	$(15, 5, 4)$
pasture (PA)	36	25	3	$(12, 12, 12)$
squash-stored (SS)	52	51	3	$(23, 21, 8)$
squash-unstored (SU)	52	52	3	$(24, 24, 4)$
tae (TA)	151	54	3	$(49, 50, 52)$
newthyroid (NT)	215	5	3	$(30, 150, 35)$
balance-scale (BS)	625	4	3	$(288, 49, 288)$
SWD (SW)	1000	10	4	$(32, 352, 399, 217)$
car (CA)	1728	21	4	$(1210, 384, 69, 65)$
bondrate (BO)	57	37	5	$(6, 33, 12, 5, 1)$
toy (TO)	300	2	5	$(35, 87, 79, 68, 31)$
eucalyptus (EU)	736	91	5	$(180, 107, 130, 214, 105)$
LEV (LE)	1000	4	5	$(93, 280, 403, 197, 27)$
automobile (AU)	205	71	6	$(3, 22, 67, 54, 32, 27)$
winequality-red (WR)	1599	11	6	$(10, 53, 681, 638, 199, 18)$
ESL (ES)	488	4	9	$(2, 12, 38, 100, 116, 135, 62, 19, 4)$
ERA (ER)	1000	4	9	$(92, 142, 181, 172, 158, 118, 88, 31, 18)$

TABLE 5
Different algorithms considered for the experiments

Abbr.	Short description
Naïve approaches	
SVC1V1	Support Vector Classifier with *OneVsOne* [59]
SVC1VA	Support Vector Classifier with *OneVsAll* [59]
SVR	Support Vector Machines for regression [52]
CSSVC	Cost-Sensitive Support Vector Classifier (CSSVC) [59]
Ordinal Binary decompositions	
SVMOP	Support Vector Machines with *OrderedPartitions* [63], [65]
NNOP	Neural Network with *OrderedPartitions* [71]
ELMOP	Extreme Learning Machine with *OrderedPartitions* [73]
Threshold models	
POM	Proportional Odds Model [80]
NNPOM	Neural Network based on Proportional Odd Model [18]
SVOREX	Support Vector Ordinal Regression with Explicit Constraints [29]
SVORIM	Support Vector Ordinal Regression with Implicit Constraints [29]
SVORLin	SVORIM using a linear kernel [29]
KDLOR	Kernel Discriminant Learning for Ordinal Regression [31]
GPOR	Gaussian Processes for Ordinal Regression [30]
REDSVM	Reduction applied to Support Vector Machines [41]
ORBALL	Ordinal Regression Boosting with All margins [107]

that they are suggested by many of the authors of the different works analysed. Starting with naïve approaches, the following methods were considered: 1) C-Support Vector Classifier (C-SVC) with *OneVsOne* and *OneVsAll* decompositions (SVC1V1 and SVC1VA), because they are the two main approaches when applying SVM to multiclass problems [59]. Although these methods consider binary decompositions, they have been included in the nominal classification group, given that they do not take the class order into account. 2) Support Vector Regression (SVR) applied to a modified dataset where the target variable $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$ is mapped to the real values $\{0, 1/(Q-1), 2/(Q-1), \ldots, 1\}$. The concrete regression model considered is the ϵ-SVR [52]. 3) Cost-Sensitive SVC (CSSVC), which is a C-SVC [59] with the *OneVsAll* decomposition, where absolute costs are included as different weights [127] for the negative class of each decomposition.

Regarding the ordinal binary decompositions, the methods considered are the following: 1) The *OrderedPartitions* decomposition was applied to the C-SVC classification algorithm (SVMOP), but including different weights, as proposed by Waegeman et al. [65]. However, given the problem of possible ambiguities recognised by the authors, probability estimates are obtained following the method presented in [128]. Then, the fusion of probabilities of Eq. (1) is performed [63]. 2) The neural network model proposed in [71] (NNOP). This model considers the *OrderedPartitions* coding scheme for the labels and a rule for decisions based on the first node whose output is higher than a predefined threshold ($T = 0.5$, in our experiments). We consider then the mean square error function over the outputs and the iRProp+ algorithm [129] to optimise the parameters. 3) Finally, the single model ordinal ELM presented in [73] (ELMOP).

The threshold models considered are the following: 1) The POM [82], with the *logit* link function (the most popular one). 2) A neural network approach based on the POM (NNPOM), similar to the one proposed by Mathieson [18]. The cross entropy function is optimised by the iRProp+ algorithm [129]. Threshold constraints are satisfied by substituting the set of parameters $\{b_1, b_2, \ldots, b_Q\}$ by $\{\alpha_1, \alpha_1 + \alpha_2^2, \ldots, \alpha_1 + \alpha_2^2 + \ldots + \alpha_Q^2\}$, which allows unconstrained optimisation of $\{\alpha_1, \ldots, \alpha_Q\}$. 3) Ordinal support vector formulations of Chu and Keerthi [29], including both explicitly and implicitly constrained alternatives (SVOREX and SVORIM). We have also included a linear version of the SVORIM method (considering the linear kernel instead of the Gaussian one) to check how the kernel trick affects the final performance (SVORLin). 4) KDLOR algorithm presented in [31]. 5) The GPOR method [30] including automatic relevance determination, as proposed by the authors. 6) The reduction from ordinal regression to binary SVM classifiers was also considered (REDSVM). The configuration used was the identity coding matrix, the absolute cost matrix and the standard binary soft-margin SVM, as proposed in [41]. 7) Finally, the ORBoost method with all margins [107] (ORBALL). As proposed by the authors, the total number of ensemble members is set to $T = 2000$, and normalised

sigmoid functions are used as the base classifier, where the smoothness parameter is $\gamma = 4$ [107].

4.1.3 Performance evaluation and model selection

Different measures can be considered for evaluating ordinal regression models [119], [130], [131]. However, the most common ones are the Mean Zero-one Error (MZE) and the Mean Absolute Error (MAE). MZE is the error rate of the classifier:

$$MZE = \frac{1}{N}\sum_{i=1}^{N}[\![y_i^* \neq y_i]\!] = 1 - Acc,$$

where y_i is the true label, y_i^* is the predicted label and Acc is the accuracy of classifier. MZE values range from 0 to 1. It is related to global performance, but without considering the order. The MAE is the average deviation in absolute value of the predicted rank ($\mathcal{O}(y_i^*)$) from the true one ($\mathcal{O}(y_i)$) [131]:

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

MAE values range from 0 to $Q-1$ (maximum deviation in number of categories). In this way, MZE considers a zero-one loss for misclassification, while MAE uses an absolute cost. We consider these costs for evaluating the datasets because they are most common (for example, see [29]–[31], just to cite some of them).

Multiple random splits of the datasets were considered. For discretised regression datasets, 20 random splits were done and the number of training and test patterns were those suggested in [30]. For real ordinal regression problems, 30 random stratified splits with 75% and 25% of the patterns in the training and test sets were considered, respectively (as suggested in [132]). All the partitions were the same for all the methods, and one model was trained and evaluated for each split. Then, MZE or MAE values were obtained, and the computational times were also gathered.

All SVM classifiers or regressors were run using the implementations available in the `libsvm` library (version 3.0) [127]. The `mnrfit` function of Matlab was used for training the POM model. The authors of GPOR, SVOREX, SVORIM, RED-SVM and ORBoost provide publicly available software implementations of their methods[3]. All the experiments were run using a common Matlab framework, with an Intel(R) Xeon(R) CPU E5405 at 2.00GHz with 8GB of RAM. This framework is available, together with all the datasets and partitions, the individual results and the detailed results of the statistical tests, on the website associated with this paper[4].

3. GPOR (http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html), SVOREX and SVORIM (http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm), ORBoost (http://www.work.caltech.edu/~htlin/program/orensemble/) and RED-SVM (http://home.caltech.edu/~htlin/program/libsvm/)

4. http://www.uco.es/grupos/ayrna/orreview

It is very important to consider a proper model selection process to assure a fair comparison. In this sense, all model hyperparameters were selected by using a nested five fold cross-validation over the training set. Once the lowest cross-validation error alternative was obtained, it was applied to the complete training set and test results were extracted. The criteria for selecting the best configuration were both MAE and MZE performances, depending on the measure we were interested in. The parameter configurations explored are now specified. The Gaussian kernel function was considered for all the kernel methods (SVC1V1, SVC1VA, SVR, CSSVC, SVMOP, SVOREX, SVORIM, REDSVM and KDLOR). The following values were considered for the width of the kernel, $\sigma \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. The cost parameter C of all SVM methods (including SVORLin) was selected within the values $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ and for the KDLOR within the values $C \in \{10^{-1}, 10^{0}, 10^{1}\}$, since, in this case, this parameter presents a different interpretation and, therefore, there is no need to use a larger spectrum of values. An additional parameter u was also needed by KDLOR, which is intended to avoid singularities in the covariance matrices. The values considered were $u \in \{10^{-6}, 10^{-5}, \ldots, 10^{-2}\}$. The range of ϵ for ϵ-SVR was $\epsilon \in \{10^{0}, 10^{1}, \ldots, 10^{3}\}$. For the neural network algorithms (NNOP and NNPOM), the number of hidden neurons, H, was selected by considering the following values, $H \in \{5, 10, 20, 30, 40\}$. The sigmoidal activation function was considered for hidden neurons. For the iRProp+ algorithm, the number of iterations, $iter$, was also decided by cross-validation, by considering the values $iter \in \{50, 100, 150, \ldots, 500\}$. The other parameters of iRProp+ were set as in [129]. For ELMOP, higher numbers of hidden neurons are considered, $H \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, given that it relies on sufficiently informative random projections [72]. With regards to the GPOR algorithm, the hyperparameters are determined by part of the optimisation process. The ORBoost process did not need any hyperparameter.

Each pair of algorithms is compared by means of the Wilcoxon test [133]. A level of significance of $\alpha = 0.1$ was considered, and the corresponding correction for the number of comparisons was also included. As 16 algorithms are compared, the total number of comparisons for each dataset is 120, so the corrected level of significance was $\alpha^* = 0.1/120 = 0.00083$.

4.2 Discretised regression datasets

Tables 6 and 7 show the results obtained for all algorithms throughout the discretised regression datasets (when considering $Q = 5$ and $Q = 10$ bins), and also the ordinal regression ones (analysed in the following subsection). The results include the average and the standard deviation of MZE and MAE, respectively. Additionally, an analysis of the ranking of each method for each dataset was done, where this ranking is 1 for the best method and 16 for the worst one. Then, the average

TABLE 6
Test MZE results for each dataset and method, including the average over all the splits and the standard deviation.

Discretised regression datasets ($Mean_{SD}$)																
Datasets	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	SVORLin	KDLOR	GPOR	REDSVM	ORBALL
P5	.546$_{.091}$	.542$_{.075}$	.492$_{.090}$	.533$_{.063}$	.521$_{.083}$	.646$_{.109}$	.652$_{.088}$	**.485$_{.118}$**	.646$_{.085}$	.508$_{.095}$	.498$_{.092}$	*.485$_{.099}$*	.512$_{.081}$	.515$_{.096}$	.506$_{.080}$	.562$_{.067}$
M5	.428$_{.060}$	.452$_{.061}$	.431$_{.056}$	.458$_{.059}$	.414$_{.063}$	.400$_{.066}$	.402$_{.074}$	*.394$_{.065}$*	.431$_{.073}$	.432$_{.058}$	.416$_{.067}$	.403$_{.061}$	.425$_{.064}$	.403$_{.054}$	.400$_{.064}$	**.389$_{.069}$**
H5	.353$_{.034}$	.374$_{.027}$	.351$_{.031}$	.372$_{.025}$	.351$_{.030}$	.358$_{.026}$	.361$_{.027}$	.355$_{.018}$	.386$_{.027}$	.329$_{.022}$	.325$_{.027}$	.350$_{.019}$	.362$_{.027}$	**.310$_{.029}$**	*.324$_{.029}$*	.328$_{.025}$
S5	*.107$_{.013}$*	.112$_{.017}$	.120$_{.016}$	.107$_{.016}$	.116$_{.020}$	.122$_{.018}$	.112$_{.014}$	.370$_{.016}$	.119$_{.015}$	.113$_{.014}$	.113$_{.014}$	.368$_{.017}$	.113$_{.013}$	.109$_{.015}$	.111$_{.018}$	**.101$_{.014}$**
A5	.512$_{.010}$	.527$_{.008}$	.530$_{.007}$	.524$_{.009}$	**.507$_{.010}$**	.520$_{.009}$	.521$_{.007}$	.539$_{.005}$	.518$_{.009}$	.512$_{.006}$	.524$_{.008}$	.541$_{.005}$	.546$_{.009}$	*.509$_{.007}$*	.524$_{.008}$	.529$_{.008}$
B5	.453$_{.064}$	.539$_{.044}$	.357$_{.056}$	.546$_{.032}$	.431$_{.035}$	.705$_{.071}$	.545$_{.046}$	*.271$_{.024}$*	.765$_{.023}$	.273$_{.025}$	.276$_{.029}$	.275$_{.021}$	.315$_{.048}$	**.267$_{.017}$**	.283$_{.038}$	.329$_{.028}$
BB5	.709$_{.021}$	.690$_{.020}$	.664$_{.020}$	.684$_{.016}$	.693$_{.019}$	.780$_{.014}$	.735$_{.026}$	*.639$_{.026}$*	.786$_{.013}$	.652$_{.024}$	.646$_{.024}$	.642$_{.023}$	.657$_{.024}$	**.585$_{.029}$**	.644$_{.024}$	.673$_{.018}$
C5	.421$_{.016}$	.470$_{.017}$	.424$_{.035}$	.463$_{.020}$	.413$_{.018}$	.426$_{.017}$	.461$_{.030}$	*.381$_{.014}$*	.453$_{.036}$	.390$_{.028}$	.397$_{.031}$	.386$_{.012}$	.429$_{.027}$	**.377$_{.014}$**	.397$_{.038}$	.458$_{.026}$
CC5	.380$_{.032}$	.425$_{.023}$	.375$_{.018}$	.421$_{.021}$	.370$_{.024}$	.406$_{.017}$	.457$_{.046}$	.342$_{.019}$	.415$_{.021}$	.343$_{.016}$	.338$_{.014}$	*.337$_{.016}$*	.387$_{.028}$	**.325$_{.017}$**	.341$_{.019}$	.434$_{.031}$
CH5	.519$_{.021}$	.542$_{.022}$	.527$_{.020}$	.546$_{.023}$	.516$_{.023}$	.520$_{.016}$	.547$_{.019}$	**.492$_{.010}$**	.527$_{.032}$	.496$_{.011}$	.499$_{.011}$	.502$_{.017}$	.510$_{.017}$	.497$_{.017}$	*.494$_{.016}$*	.515$_{.012}$
CE5	.564$_{.018}$	.576$_{.016}$	.587$_{.015}$	.587$_{.021}$	.567$_{.017}$	.587$_{.023}$	.595$_{.019}$	.558$_{.010}$	.596$_{.023}$	*.555$_{.013}$*	.556$_{.012}$	.567$_{.016}$	.568$_{.018}$	**.535$_{.009}$**	.559$_{.013}$	.592$_{.016}$
CEE5	.564$_{.016}$	.585$_{.017}$	.573$_{.016}$	.585$_{.018}$	.560$_{.020}$	.600$_{.016}$	.599$_{.016}$	.584$_{.017}$	.634$_{.020}$	.553$_{.018}$	.555$_{.016}$	.585$_{.020}$	*.552$_{.016}$*	**.539$_{.009}$**	.553$_{.013}$	.583$_{.011}$
R_{D5}	8.5000	12.1667	9.6667	12.0000	7.8333	11.6250	12.4167	5.7500	13.3750	5.3750	5.2917	6.8333	9.0833	**2.6667**	*4.5000*	8.9167
P10	.775$_{.069}$	.758$_{.064}$	.769$_{.091}$	.758$_{.063}$	.763$_{.079}$	.802$_{.075}$	.794$_{.081}$	.719$_{.076}$	.850$_{.073}$	.783$_{.076}$	.735$_{.095}$	**.708$_{.078}$**	.756$_{.086}$	.779$_{.087}$	*.717$_{.102}$*	.760$_{.091}$
M10	**.597$_{.055}$**	.641$_{.062}$	.634$_{.065}$	.634$_{.047}$	.643$_{.047}$	.621$_{.058}$	.610$_{.054}$	.642$_{.059}$	.653$_{.054}$	.624$_{.062}$	.629$_{.054}$	.648$_{.061}$	.651$_{.065}$	.642$_{.056}$	.618$_{.051}$	*.606$_{.055}$*
H10	.597$_{.025}$	.636$_{.028}$	.590$_{.035}$	.623$_{.019}$	.589$_{.031}$	.584$_{.033}$	.599$_{.029}$	.592$_{.032}$	.606$_{.033}$	.574$_{.033}$	.557$_{.029}$	.602$_{.026}$	.599$_{.038}$	**.542$_{.026}$**	*.548$_{.032}$*	.551$_{.029}$
S10	**.206$_{.022}$**	.218$_{.019}$	.231$_{.019}$	.215$_{.020}$	*.212$_{.021}$*	.249$_{.033}$	.238$_{.018}$	.561$_{.019}$	.257$_{.025}$	.235$_{.023}$	.234$_{.021}$	.589$_{.022}$	.230$_{.019}$	.249$_{.019}$	.230$_{.021}$	.232$_{.025}$
A10	.713$_{.008}$	.733$_{.012}$	.739$_{.005}$	.725$_{.008}$	.718$_{.008}$	.725$_{.010}$	.731$_{.008}$	.743$_{.005}$	.719$_{.008}$	*.709$_{.008}$*	.733$_{.008}$	.748$_{.007}$	.744$_{.009}$	**.708$_{.005}$**	.728$_{.010}$	.732$_{.007}$
B10	.755$_{.033}$	.766$_{.020}$	.588$_{.052}$	.763$_{.020}$	.704$_{.047}$	.851$_{.044}$	.754$_{.044}$	**.472$_{.019}$**	.881$_{.013}$	.506$_{.020}$	.494$_{.021}$	*.483$_{.023}$*	.524$_{.047}$	.485$_{.016}$	.487$_{.022}$	.560$_{.023}$
BB10	.858$_{.010}$	.860$_{.011}$	.825$_{.024}$	.848$_{.009}$	.861$_{.015}$	.895$_{.006}$	.869$_{.014}$	.805$_{.014}$	.894$_{.006}$	.814$_{.017}$	.811$_{.018}$	*.804$_{.017}$*	.816$_{.019}$	**.777$_{.014}$**	.815$_{.017}$	.831$_{.012}$
C10	.643$_{.016}$	.677$_{.019}$	.634$_{.032}$	.677$_{.016}$	.631$_{.016}$	.634$_{.011}$	.672$_{.024}$	**.580$_{.014}$**	.657$_{.035}$	.598$_{.027}$	.607$_{.024}$	.589$_{.017}$	.644$_{.028}$	*.588$_{.017}$*	.601$_{.027}$	.654$_{.015}$
CC10	.603$_{.013}$	.647$_{.020}$	.585$_{.032}$	.651$_{.027}$	.591$_{.019}$	.605$_{.021}$	.645$_{.024}$	*.534$_{.012}$*	.628$_{.021}$	.549$_{.030}$	.546$_{.021}$	.536$_{.017}$	.611$_{.034}$	**.527$_{.013}$**	.548$_{.020}$	.642$_{.024}$
CH10	.732$_{.019}$	.744$_{.016}$	.725$_{.013}$	.744$_{.013}$	.733$_{.022}$	.723$_{.015}$	.746$_{.017}$	.704$_{.011}$	.719$_{.022}$	*.702$_{.012}$*	.707$_{.015}$	.710$_{.012}$	.717$_{.018}$	**.696$_{.014}$**	.706$_{.018}$	.713$_{.011}$
CE10	.770$_{.012}$	.780$_{.015}$	.770$_{.009}$	.779$_{.011}$	.768$_{.019}$	.778$_{.013}$	.778$_{.012}$	*.753$_{.012}$*	.786$_{.014}$	.755$_{.014}$	.756$_{.014}$	.763$_{.016}$	.776$_{.014}$	**.736$_{.007}$**	.758$_{.018}$	.778$_{.010}$
CEE10	.778$_{.022}$	.782$_{.014}$	.767$_{.014}$	.782$_{.013}$	.771$_{.013}$	.785$_{.009}$	.788$_{.009}$	.780$_{.008}$	.813$_{.012}$	*.758$_{.009}$*	.759$_{.012}$	.780$_{.009}$	.761$_{.015}$	**.749$_{.010}$**	.762$_{.018}$	.774$_{.007}$
R_{D10}	8.0833	12.2500	8.6667	11.3333	8.3333	10.7500	12.0833	6.1667	13.5000	5.3333	5.5000	7.7500	9.2500	**4.0833**	*4.5833*	8.3333
R_{D}	8.2917	12.2083	9.1667	11.6667	8.0833	11.1875	12.2500	5.9583	13.4375	5.3542	5.3958	7.2917	9.1667	**3.3750**	*4.5417*	8.6250
Ordinal regression datasets ($Mean_{SD}$)																
Datasets	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	SVORLin	KDLOR	GPOR	REDSVM	ORBALL
CL	*.278$_{.126}$*	.278$_{.110}$	.317$_{.080}$	**.261$_{.136}$**	.367$_{.102}$	.283$_{.125}$	.422$_{.218}$	.383$_{.170}$	.356$_{.143}$	.356$_{.129}$	.383$_{.117}$	.361$_{.099}$	.339$_{.155}$	.394$_{.093}$	.372$_{.121}$	.356$_{.129}$
PA	.307$_{.119}$	.330$_{.107}$	.337$_{.129}$	.322$_{.114}$	.319$_{.091}$	**.237$_{.116}$**	.400$_{.184}$	.504$_{.154}$	.344$_{.178}$	.348$_{.119}$	.344$_{.122}$	.344$_{.122}$	.326$_{.116}$	.478$_{.178}$	.326$_{.116}$	*.300$_{.121}$*
SS	**.359$_{.095}$**	.403$_{.145}$	.387$_{.113}$	.395$_{.134}$	.403$_{.094}$	.395$_{.116}$	.444$_{.162}$	.618$_{.152}$	.500$_{.131}$	.374$_{.127}$	.372$_{.129}$	.372$_{.133}$	.392$_{.125}$	.549$_{.100}$	.379$_{.132}$	*.364$_{.124}$*
SU	**.221$_{.121}$**	.279$_{.158}$	*.244$_{.103}$*	.269$_{.156}$	.267$_{.116}$	.295$_{.113}$	.387$_{.129}$	.651$_{.142}$	.390$_{.143}$	.264$_{.114}$	.269$_{.114}$	.315$_{.113}$	.264$_{.108}$	.356$_{.162}$	.269$_{.119}$	.297$_{.098}$
TA	.439$_{.059}$	.448$_{.065}$	**.396$_{.073}$**	.428$_{.072}$	.456$_{.057}$	.415$_{.060}$	.436$_{.079}$	.496$_{.077}$	.453$_{.090}$	.411$_{.069}$	.401$_{.072}$	.460$_{.072}$	.433$_{.055}$	.672$_{.041}$	*.399$_{.068}$*	.403$_{.057}$
NT	.035$_{.027}$	.041$_{.029}$	.044$_{.024}$	.038$_{.025}$	.041$_{.027}$	.035$_{.022}$	.057$_{.024}$	*.028$_{.022}$*	.033$_{.025}$	.034$_{.023}$	.034$_{.023}$	.031$_{.024}$	**.026$_{.020}$**	.034$_{.024}$	.032$_{.023}$	.042$_{.029}$
BS	.028$_{.015}$	.033$_{.012}$	.165$_{.027}$	.031$_{.013}$	.034$_{.014}$	.039$_{.013}$	.087$_{.023}$	.094$_{.019}$	.062$_{.048}$	*.002$_{.006}$*	.002$_{.006}$	.094$_{.019}$	.160$_{.031}$	.034$_{.012}$	**.001$_{.004}$**	.032$_{.016}$
SW	*.422$_{.034}$*	.437$_{.035}$	.435$_{.030}$	.429$_{.032}$	.424$_{.035}$	.423$_{.035}$	.426$_{.025}$	.432$_{.030}$	.456$_{.032}$	.432$_{.030}$	.431$_{.027}$	.431$_{.030}$	.514$_{.028}$	**.422$_{.031}$**	.429$_{.027}$	.439$_{.032}$
CA	*.006$_{.005}$*	.014$_{.006}$	.027$_{.006}$	.014$_{.006}$	**.003$_{.004}$**	.026$_{.013}$	.159$_{.015}$	.843$_{.306}$	.106$_{.022}$	.012$_{.006}$	.012$_{.005}$	.077$_{.010}$	.047$_{.010}$	.037$_{.009}$	.012$_{.004}$	.012$_{.006}$
BO	*.429$_{.042}$*	.431$_{.060}$	.458$_{.072}$	.433$_{.060}$	.447$_{.098}$	.533$_{.093}$	.567$_{.135}$	.656$_{.161}$	.618$_{.143}$	.453$_{.062}$	.451$_{.074}$	.433$_{.085}$	.473$_{.093}$	**.422$_{.032}$**	.436$_{.055}$	.458$_{.091}$
TO	.050$_{.026}$	.055$_{.028}$	.068$_{.037}$	.054$_{.028}$	.072$_{.028}$	.062$_{.031}$	.071$_{.029}$	.711$_{.026}$	.062$_{.032}$	**.020$_{.014}$**	*.020$_{.014}$*	.730$_{.016}$	.107$_{.032}$	.046$_{.022}$	.023$_{.012}$	.052$_{.025}$
EU	.360$_{.029}$	.445$_{.026}$	.361$_{.027}$	.435$_{.026}$	*.350$_{.026}$*	.423$_{.036}$	.428$_{.029}$	.851$_{.016}$	.462$_{.042}$	.364$_{.027}$	.361$_{.033}$	.357$_{.023}$	.372$_{.028}$	**.314$_{.034}$**	.362$_{.035}$	.380$_{.029}$
LE	.368$_{.025}$	.372$_{.020}$	.375$_{.025}$	**.367$_{.019}$**	*.367$_{.025}$*	.375$_{.026}$	.367$_{.022}$	.377$_{.028}$	.382$_{.024}$	.375$_{.022}$	.380$_{.018}$	.384$_{.028}$	.458$_{.031}$	.388$_{.030}$	.373$_{.024}$	.391$_{.029}$
AU	**.246$_{.058}$**	.260$_{.058}$	.321$_{.067}$	.265$_{.061}$	*.258$_{.042}$*	.389$_{.060}$	.381$_{.061}$	.533$_{.194}$	.550$_{.077}$	.316$_{.055}$	.323$_{.069}$	.408$_{.068}$	.301$_{.067}$	.389$_{.073}$	.317$_{.070}$	.294$_{.055}$
WI	.352$_{.023}$	.361$_{.024}$	.372$_{.019}$	.362$_{.023}$	.358$_{.020}$	.402$_{.018}$	.397$_{.016}$	.403$_{.015}$	.401$_{.022}$	.373$_{.019}$	.373$_{.020}$	.407$_{.015}$	*.350$_{.019}$*	.394$_{.015}$	.373$_{.020}$	**.334$_{.021}$**
ES	.307$_{.037}$	.328$_{.027}$	.296$_{.034}$	.320$_{.026}$	.289$_{.032}$	.308$_{.039}$	.302$_{.039}$	.295$_{.034}$	.341$_{.129}$	.290$_{.026}$	**.284$_{.032}$**	.293$_{.033}$	.355$_{.034}$	*.287$_{.031}$*	.287$_{.030}$	.323$_{.022}$
ER	.736$_{.025}$	.825$_{.026}$	.750$_{.023}$	.801$_{.029}$	.745$_{.023}$	**.708$_{.017}$**	.746$_{.019}$	.744$_{.021}$	.727$_{.028}$	.714$_{.026}$	.751$_{.021}$	.750$_{.023}$	.805$_{.031}$	*.712$_{.027}$*	.751$_{.019}$	.760$_{.021}$
R_{OR}	**4.0000**	8.8824	8.5588	7.1471	6.4706	8.3529	11.4706	12.9118	12.1765	6.6176	7.0588	9.9706	9.7647	8.7059	*5.8824*	8.0294

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

values of these rankings were obtained for discretised regression problems of 5 classes (R_{D5}), of 10 classes (R_{D10}), for all discretised regression problems (R_{D}) and for real ordinal regression ones (R_{OR}), as a reference of the general performance. For all methods including a model selection process, Table 6 shows the results when using MZE as the selection criterion, while the results in Table 7 are obtained using MAE for this selection. In addition, the average rankings of the total computational time (the sum of cross-validation, training and test times) are presented in Table 8. For all these Tables, the best method for each dataset (or set of datasets) is highlighted in bold face and the second one in italics.

As previously stated, the Wilcoxon test was applied to check the existence of significant differences. Using this test, each pair of methods was compared for each dataset and the total number of statistically significant wins or losses was recorded, together with the number of draws (or absence of statistically significant differences). These results are included in Table 9 for the discretised regression datasets ($Q = 5$ and $Q = 10$). The number of wins (w), draws (d) and losses (l) for $15 \times 24 = 360$ comparisons are included (24 datasets and 15 methods to compare each method against). The methods are ordered by the number of statistically significant wins.

By analysing Tables 6, 7, 8, and specially Table 9, the best performing ordinal regression methods from the different families in the taxonomy can be obtained. From the naïve approaches, SVC1V1 obtains better accuracy or MZE while SVR is better on MAE. However, both improved performances imply worse results for MAE and MZE, respectively. The computational time of SVR is higher, given that an additional parameter has to be cross-validated. In general, SVC1VA and CSSVC show worse MAE and MZE than SVC1V1. When considering ordinal binary decompositions, SVMOP is

TABLE 7
Test MAE results for each dataset and method, including the average over all the splits and the standard deviation.

Discretised regression datasets ($Mean_{SD}$)																
Datasets	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	SVORLin	KDLOR	GPOR	REDSVM	ORBALL
P5	$0.79_{0.13}$	$0.75_{0.18}$	$0.66_{0.14}$	$0.74_{0.15}$	$0.69_{0.16}$	$0.91_{0.21}$	$1.12_{0.26}$	$0.70_{0.20}$	$0.96_{0.21}$	$0.66_{0.12}$	$\mathbf{0.63_{0.11}}$	$0.64_{0.14}$	$0.67_{0.19}$	$0.65_{0.13}$	*0.64$_{0.14}$*	$0.73_{0.10}$
M5	$0.47_{0.08}$	$0.55_{0.08}$	$0.48_{0.09}$	$0.55_{0.10}$	$0.46_{0.08}$	$0.46_{0.07}$	$0.45_{0.09}$	*0.43$_{0.08}$*	$0.49_{0.07}$	$0.47_{0.07}$	$0.44_{0.10}$	$0.43_{0.07}$	$0.49_{0.10}$	$0.44_{0.07}$	$0.46_{0.09}$	$\mathbf{0.41_{0.08}}$
H5	$0.41_{0.05}$	$0.44_{0.04}$	$0.38_{0.04}$	$0.44_{0.04}$	$0.40_{0.04}$	$0.41_{0.03}$	$0.42_{0.04}$	$0.40_{0.02}$	$0.45_{0.05}$	$0.36_{0.03}$	$0.36_{0.03}$	$0.40_{0.02}$	$0.39_{0.05}$	$\mathbf{0.34_{0.04}}$	*0.36$_{0.03}$*	$0.36_{0.03}$
S5	$0.11_{0.01}$	$0.11_{0.01}$	$0.12_{0.02}$	*0.11$_{0.02}$*	$0.11_{0.02}$	$0.13_{0.02}$	$0.11_{0.01}$	$0.39_{0.02}$	$0.12_{0.02}$	$0.11_{0.01}$	$0.11_{0.02}$	$0.39_{0.02}$	$0.11_{0.01}$	$0.11_{0.01}$	$0.11_{0.02}$	$\mathbf{0.10_{0.01}}$
A5	$0.71_{0.02}$	$0.79_{0.02}$	$0.66_{0.01}$	$0.78_{0.02}$	$0.67_{0.01}$	$0.66_{0.01}$	$0.67_{0.01}$	$0.69_{0.01}$	$0.70_{0.01}$	$0.66_{0.01}$	$\mathbf{0.65_{0.01}}$	$0.69_{0.01}$	$0.76_{0.02}$	$0.68_{0.01}$	*0.65$_{0.01}$*	$0.67_{0.01}$
B5	$0.50_{0.08}$	$0.67_{0.09}$	$0.36_{0.05}$	$0.67_{0.07}$	$0.47_{0.04}$	$1.13_{0.19}$	$0.69_{0.10}$	*0.28$_{0.03}$*	$1.32_{0.19}$	$0.28_{0.03}$	$0.28_{0.03}$	$0.28_{0.03}$	$0.33_{0.06}$	$\mathbf{0.27_{0.02}}$	$0.28_{0.03}$	$0.34_{0.03}$
BB5	$1.18_{0.07}$	$1.18_{0.05}$	$0.94_{0.08}$	$1.19_{0.07}$	$1.09_{0.07}$	$1.40_{0.06}$	$1.30_{0.15}$	$0.98_{0.09}$	$1.47_{0.09}$	$0.94_{0.08}$	$0.92_{0.07}$	$0.93_{0.07}$	$0.95_{0.08}$	$\mathbf{0.83_{0.08}}$	$0.93_{0.07}$	*0.89$_{0.05}$*
C5	$0.51_{0.03}$	$0.59_{0.05}$	$0.49_{0.03}$	$0.57_{0.04}$	$0.50_{0.03}$	$0.50_{0.03}$	$0.55_{0.04}$	*0.43$_{0.02}$*	$0.56_{0.06}$	$0.45_{0.05}$	$0.45_{0.04}$	$0.44_{0.01}$	$0.52_{0.06}$	$\mathbf{0.43_{0.02}}$	$0.44_{0.04}$	$0.52_{0.04}$
CC5	$0.43_{0.05}$	$0.50_{0.03}$	$0.40_{0.02}$	$0.50_{0.03}$	$0.42_{0.04}$	$0.49_{0.20}$	$0.53_{0.06}$	$0.38_{0.03}$	$0.48_{0.03}$	$0.37_{0.02}$	$0.37_{0.01}$	*0.37$_{0.02}$*	$0.44_{0.04}$	$\mathbf{0.35_{0.02}}$	$0.37_{0.02}$	$0.48_{0.05}$
CH5	$0.66_{0.04}$	$0.75_{0.04}$	$0.63_{0.03}$	$0.75_{0.04}$	$0.64_{0.03}$	$0.65_{0.03}$	$0.70_{0.03}$	*0.59$_{0.01}$*	$0.67_{0.05}$	$0.60_{0.02}$	$0.60_{0.02}$	$0.60_{0.02}$	$0.65_{0.04}$	$0.62_{0.03}$	$\mathbf{0.59_{0.02}}$	$0.64_{0.02}$
CE5	$0.79_{0.03}$	$0.86_{0.04}$	$0.76_{0.03}$	$0.87_{0.05}$	$0.77_{0.03}$	$0.81_{0.03}$	$0.80_{0.03}$	$0.73_{0.02}$	$0.87_{0.05}$	$0.74_{0.03}$	$\mathbf{0.73_{0.02}}$	$0.73_{0.02}$	$0.79_{0.04}$	$0.74_{0.02}$	*0.73$_{0.02}$*	$0.79_{0.03}$
CEE5	$0.81_{0.03}$	$0.88_{0.08}$	$0.74_{0.02}$	$0.86_{0.04}$	$0.79_{0.04}$	$0.87_{0.04}$	$0.83_{0.03}$	$0.78_{0.03}$	$0.93_{0.04}$	$0.74_{0.03}$	*0.72$_{0.03}$*	$0.78_{0.03}$	$0.74_{0.03}$	$0.74_{0.02}$	$\mathbf{0.72_{0.03}}$	$0.73_{0.01}$
R_{D5}	10.7500	13.9167	7.3333	12.9167	8.7500	11.8333	12.2500	6.7500	14.3333	5.6667	**3.2500**	5.8333	8.9167	3.5833	*3.5000*	6.4167
P10	$1.88_{0.40}$	$1.88_{0.35}$	$1.43_{0.22}$	$1.82_{0.43}$	$1.46_{0.30}$	$1.77_{0.37}$	$2.08_{0.53}$	$1.54_{0.29}$	$2.20_{0.80}$	$1.44_{0.23}$	*1.39$_{0.19}$*	$1.47_{0.24}$	$1.43_{0.26}$	$1.55_{0.24}$	$\mathbf{1.32_{0.19}}$	$1.53_{0.22}$
M10	$1.05_{0.14}$	$1.14_{0.15}$	$1.03_{0.15}$	$1.11_{0.16}$	$0.99_{0.09}$	$0.94_{0.16}$	$0.92_{0.12}$	$0.92_{0.11}$	$1.12_{0.15}$	$0.96_{0.12}$	$0.92_{0.13}$	$\mathbf{0.91_{0.11}}$	$1.09_{0.20}$	$0.97_{0.11}$	$0.92_{0.11}$	*0.91$_{0.12}$*
H10	$0.89_{0.08}$	$1.07_{0.09}$	$0.80_{0.06}$	$1.03_{0.09}$	$0.88_{0.06}$	$0.87_{0.07}$	$0.90_{0.06}$	$0.88_{0.05}$	$0.95_{0.11}$	$0.78_{0.06}$	$0.77_{0.06}$	$0.89_{0.05}$	$0.84_{0.06}$	$\mathbf{0.76_{0.05}}$	$0.76_{0.07}$	*0.76$_{0.06}$*
S10	$\mathbf{0.22_{0.02}}$	$0.22_{0.02}$	$0.24_{0.02}$	$0.23_{0.02}$	*0.22$_{0.02}$*	$0.27_{0.04}$	$0.25_{0.02}$	$0.82_{0.03}$	$0.27_{0.03}$	$0.23_{0.02}$	$0.24_{0.02}$	$0.80_{0.04}$	$0.24_{0.02}$	$0.25_{0.02}$	$0.23_{0.02}$	$0.24_{0.02}$
A10	$1.56_{0.03}$	$1.73_{0.06}$	$1.38_{0.02}$	$1.74_{0.03}$	$1.43_{0.03}$	$1.38_{0.02}$	$1.39_{0.02}$	$1.45_{0.01}$	$1.49_{0.02}$	$1.47_{0.02}$	*1.36$_{0.02}$*	$1.44_{0.02}$	$1.64_{0.04}$	$1.49_{0.02}$	$\mathbf{1.36_{0.02}}$	$1.40_{0.02}$
B10	$1.47_{0.28}$	$1.62_{0.09}$	$0.72_{0.11}$	$1.64_{0.16}$	$0.98_{0.06}$	$2.24_{0.58}$	$1.46_{0.29}$	$\mathbf{0.54_{0.03}}$	$2.87_{0.43}$	$0.59_{0.05}$	$0.57_{0.03}$	*0.56$_{0.03}$*	$0.63_{0.10}$	$0.57_{0.03}$	$0.56_{0.05}$	$0.67_{0.04}$
BB10	$2.51_{0.10}$	$2.60_{0.18}$	$1.91_{0.12}$	$2.58_{0.14}$	$2.27_{0.19}$	$2.91_{0.10}$	$2.56_{0.30}$	$1.98_{0.14}$	$3.00_{0.13}$	$1.92_{0.14}$	$1.88_{0.13}$	$1.92_{0.17}$	$1.92_{0.12}$	*1.83$_{0.18}$*	$1.88_{0.13}$	$\mathbf{1.83_{0.10}}$
C10	$1.15_{0.06}$	$1.30_{0.11}$	$0.99_{0.05}$	$1.28_{0.13}$	$1.06_{0.05}$	$1.06_{0.06}$	$1.18_{0.06}$	$\mathbf{0.87_{0.03}}$	$1.16_{0.10}$	$0.93_{0.05}$	$0.90_{0.03}$	*0.88$_{0.03}$*	$1.08_{0.10}$	$0.92_{0.04}$	$0.90_{0.04}$	$1.04_{0.06}$
CC10	$0.94_{0.05}$	$1.14_{0.06}$	$0.83_{0.05}$	$1.12_{0.07}$	$0.89_{0.05}$	$0.92_{0.05}$	$1.04_{0.10}$	$0.75_{0.04}$	$1.05_{0.12}$	$0.79_{0.06}$	$0.76_{0.05}$	*0.75$_{0.03}$*	$0.97_{0.10}$	$\mathbf{0.74_{0.03}}$	$0.76_{0.03}$	$0.96_{0.09}$
CH10	$1.46_{0.06}$	$1.67_{0.08}$	$1.27_{0.04}$	$1.63_{0.07}$	$1.37_{0.06}$	$1.34_{0.06}$	$1.43_{0.05}$	$\mathbf{1.23_{0.04}}$	$1.40_{0.11}$	$1.31_{0.05}$	$1.25_{0.04}$	*1.24$_{0.04}$*	$1.41_{0.10}$	$1.30_{0.06}$	$1.25_{0.04}$	$1.31_{0.03}$
CE10	$1.77_{0.09}$	$1.91_{0.12}$	$1.55_{0.04}$	$1.92_{0.17}$	$1.64_{0.06}$	$1.72_{0.07}$	$1.65_{0.06}$	$1.54_{0.06}$	$1.86_{0.11}$	$1.65_{0.07}$	*1.51$_{0.04}$*	$1.53_{0.06}$	$1.75_{0.12}$	$1.61_{0.06}$	$\mathbf{1.50_{0.05}}$	$1.64_{0.07}$
CEE10	$1.76_{0.05}$	$1.92_{0.06}$	$1.53_{0.05}$	$1.89_{0.06}$	$1.64_{0.06}$	$1.77_{0.07}$	$1.72_{0.06}$	$1.64_{0.04}$	$2.00_{0.14}$	$1.62_{0.08}$	$\mathbf{1.50_{0.04}}$	$1.63_{0.05}$	$1.61_{0.09}$	$1.64_{0.08}$	*1.51$_{0.06}$*	$1.51_{0.03}$
R_{D10}	11.1667	14.0833	6.0833	13.5833	8.2500	10.3333	10.8333	6.2500	14.1667	6.9167	*3.8333*	5.7500	9.1667	6.4167	**3.0833**	6.0833
R_{D}	10.9583	14.0000	6.7083	13.2500	8.5000	11.0833	11.5417	6.5000	14.2500	6.2917	*3.5417*	5.7917	9.0417	5.0000	**3.2917**	6.2500
Ordinal regression datasets ($Mean_{SD}$)																
Datasets	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	SVORLin	KDLOR	GPOR	REDSVM	ORBALL
CL	$0.52_{0.22}$	$0.46_{0.19}$	$\mathbf{0.38_{0.15}}$	$0.46_{0.23}$	$0.50_{0.18}$	$0.46_{0.25}$	$0.52_{0.28}$	$0.53_{0.25}$	$0.48_{0.23}$	$0.48_{0.13}$	$0.52_{0.17}$	$0.46_{0.16}$	$0.52_{0.22}$	$0.51_{0.17}$	$0.46_{0.16}$	*0.42$_{0.18}$*
PA	$0.30_{0.14}$	$0.32_{0.10}$	$0.32_{0.12}$	$0.34_{0.13}$	*0.29$_{0.13}$*	$\mathbf{0.24_{0.11}}$	$0.40_{0.14}$	$0.59_{0.20}$	$0.37_{0.22}$	$0.33_{0.12}$	$0.34_{0.12}$	$0.34_{0.11}$	$0.34_{0.13}$	$0.49_{0.19}$	$0.33_{0.11}$	$0.30_{0.12}$
SS	$0.38_{0.14}$	$0.46_{0.16}$	$0.37_{0.12}$	$0.43_{0.15}$	$0.41_{0.12}$	$0.42_{0.13}$	$0.48_{0.18}$	$0.81_{0.25}$	$0.54_{0.16}$	$0.37_{0.15}$	$0.38_{0.13}$	$0.38_{0.14}$	$0.37_{0.15}$	$0.63_{0.15}$	$\mathbf{0.35_{0.15}}$	*0.36$_{0.12}$*
SU	$\mathbf{0.22_{0.13}}$	$0.28_{0.18}$	$0.27_{0.10}$	$0.28_{0.17}$	$0.27_{0.12}$	$0.28_{0.10}$	$0.42_{0.14}$	$0.83_{0.23}$	$0.42_{0.14}$	$0.26_{0.12}$	$0.27_{0.11}$	$0.32_{0.11}$	*0.25$_{0.13}$*	$0.36_{0.16}$	$0.26_{0.12}$	$0.31_{0.11}$
TA	$0.54_{0.10}$	$0.50_{0.09}$	$0.48_{0.07}$	$0.52_{0.11}$	$0.50_{0.08}$	$0.54_{0.11}$	$0.62_{0.12}$	$0.63_{0.12}$	$0.58_{0.13}$	$0.47_{0.06}$	$0.47_{0.07}$	$0.55_{0.08}$	$\mathbf{0.46_{0.07}}$	$0.86_{0.16}$	*0.46$_{0.06}$*	$0.50_{0.09}$
NT	$0.04_{0.03}$	$0.04_{0.02}$	$0.05_{0.03}$	$0.04_{0.02}$	$0.04_{0.03}$	$0.04_{0.02}$	$0.05_{0.02}$	*0.03$_{0.02}$*	$0.03_{0.03}$	$0.03_{0.02}$	$0.03_{0.02}$	$0.03_{0.02}$	$\mathbf{0.02_{0.02}}$	$0.03_{0.02}$	$0.03_{0.02}$	$0.04_{0.03}$
BS	$0.03_{0.01}$	$0.03_{0.01}$	$0.17_{0.03}$	$0.03_{0.01}$	$0.03_{0.02}$	$0.04_{0.01}$	$0.09_{0.03}$	$0.11_{0.02}$	$0.11_{0.19}$	*0.00$_{0.01}$*	$0.00_{0.01}$	$0.11_{0.02}$	$0.16_{0.02}$	$0.03_{0.01}$	$\mathbf{0.00_{0.00}}$	$0.03_{0.02}$
SW	$\mathbf{0.44_{0.04}}$	$0.48_{0.04}$	$0.45_{0.03}$	$0.48_{0.04}$	$0.45_{0.04}$	$0.45_{0.04}$	$0.45_{0.03}$	$0.45_{0.03}$	$0.48_{0.04}$	$0.45_{0.03}$	$0.45_{0.03}$	$0.45_{0.03}$	$0.58_{0.03}$	*0.44$_{0.03}$*	$0.45_{0.03}$	$0.46_{0.04}$
CA	*0.01$_{0.01}$*	$0.01_{0.01}$	$0.03_{0.01}$	$0.02_{0.01}$	$\mathbf{0.00_{0.00}}$	$0.03_{0.01}$	$0.18_{0.01}$	$1.45_{0.55}$	$0.12_{0.03}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.08_{0.01}$	$0.05_{0.01}$	$0.04_{0.01}$	$0.01_{0.00}$	$0.01_{0.01}$
BO	$0.64_{0.10}$	$0.60_{0.10}$	$0.59_{0.10}$	*0.57$_{0.11}$*	$0.59_{0.12}$	$0.67_{0.16}$	$0.65_{0.17}$	$0.95_{0.32}$	$0.80_{0.21}$	$0.62_{0.09}$	$0.61_{0.09}$	$0.60_{0.10}$	$0.63_{0.08}$	$0.62_{0.06}$	$0.61_{0.08}$	$\mathbf{0.53_{0.11}}$
TO	$0.05_{0.02}$	$0.06_{0.03}$	$0.06_{0.03}$	$0.06_{0.03}$	$0.07_{0.03}$	$0.06_{0.03}$	$0.08_{0.03}$	$0.98_{0.04}$	$0.06_{0.03}$	$\mathbf{0.02_{0.01}}$	*0.02$_{0.01}$*	$0.96_{0.07}$	$0.11_{0.03}$	$0.05_{0.02}$	$0.02_{0.01}$	$0.05_{0.02}$
EU	$0.41_{0.04}$	$0.51_{0.04}$	$0.40_{0.03}$	$0.51_{0.04}$	$0.40_{0.03}$	$0.48_{0.04}$	$0.53_{0.05}$	$1.94_{0.25}$	$0.57_{0.08}$	$0.40_{0.04}$	$0.39_{0.03}$	*0.38$_{0.03}$*	$0.40_{0.03}$	$\mathbf{0.33_{0.04}}$	$0.40_{0.04}$	$0.41_{0.04}$
LE	*0.40$_{0.03}$*	$0.41_{0.03}$	$0.41_{0.02}$	$0.41_{0.03}$	$\mathbf{0.40_{0.03}}$	$0.41_{0.03}$	$0.41_{0.03}$	$0.41_{0.03}$	$0.42_{0.03}$	$0.41_{0.02}$	$0.41_{0.02}$	$0.41_{0.03}$	$0.51_{0.04}$	$0.42_{0.03}$	$0.41_{0.02}$	$0.43_{0.03}$
AU	*0.36$_{0.10}$*	$0.39_{0.10}$	$0.39_{0.08}$	$0.39_{0.10}$	$0.37_{0.08}$	$0.50_{0.09}$	$0.54_{0.09}$	$0.95_{0.69}$	$0.85_{0.15}$	$0.42_{0.09}$	$0.39_{0.08}$	$0.47_{0.09}$	$0.39_{0.08}$	$0.59_{0.13}$	$0.40_{0.09}$	$\mathbf{0.35_{0.08}}$
WI	$0.41_{0.02}$	$0.42_{0.02}$	$0.42_{0.02}$	$0.42_{0.02}$	$0.41_{0.02}$	$0.44_{0.02}$	$0.43_{0.02}$	$0.44_{0.02}$	$0.45_{0.03}$	$0.42_{0.02}$	$0.42_{0.02}$	$0.44_{0.02}$	*0.39$_{0.02}$*	$0.42_{0.02}$	$0.42_{0.02}$	$\mathbf{0.37_{0.02}}$
ES	$0.32_{0.04}$	$0.35_{0.04}$	$0.31_{0.04}$	$0.34_{0.03}$	$0.30_{0.04}$	$0.32_{0.04}$	$0.32_{0.03}$	$0.31_{0.04}$	$0.46_{0.63}$	$0.30_{0.04}$	$\mathbf{0.30_{0.03}}$	$0.31_{0.03}$	$0.37_{0.04}$	*0.30$_{0.03}$*	$0.31_{0.04}$	$0.34_{0.03}$
ER	$1.29_{0.07}$	$2.02_{0.12}$	$1.22_{0.05}$	$1.91_{0.15}$	$1.24_{0.04}$	$\mathbf{1.18_{0.02}}$	$1.24_{0.04}$	$1.22_{0.05}$	$1.26_{0.06}$	*1.21$_{0.06}$*	$1.21_{0.04}$	$1.22_{0.04}$	$1.78_{0.10}$	$1.24_{0.05}$	$1.22_{0.04}$	$1.25_{0.04}$
R_{OR}	6.4412	9.2941	7.0588	8.6471	5.8824	8.4706	12.5000	12.5000	13.1765	*5.4118*	6.4706	9.2647	9.2941	9.6471	**4.8235**	7.1176

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

the best performing one, improving MAE and MZE with respect to NNOP and ELMOP methods. ELMOP is slightly better than NNOP in MAE, but the opposite happens when observing MZE. However, ELMOP is clearly the fastest ordinal binary decomposition method. From the threshold methods, it is clear that GPOR is the best performing one in MZE, REDSVM obtains the lowest MAE (although with higher computational cost), and the POM is the fastest one, followed by SVOREX. SVORLin obtains very good MAE results, close to that of SVORIM. Although one could have expected a lower computational time for SVORLin, the pressure of obtaining a low error rate using a linear model when $C = 1000$ makes convergence very difficult.

4.3 Ordinal regression datasets

This section presents the study performed on the ordinal regression datasets. The objective is to analyse how the methods perform in more realistic situations, where the underlying variable is really unobservable and traditional classification problems appear (e.g. class imbalance). Tables 6, 7 and 8 show the complete set of results obtained, and Table 10 includes the Wilcoxon tests. Its format is similar to that in Table 9, but the number of comparisons is now $15 \times 17 = 255$. In general, similar conclusions can now be obtained, but there are some differences. SVC1V1 performance regarding MZE is now better, when compared to the rest of methods. The problems associated with these datasets (mainly uneven distribution ratios) are generally better solved by using this kind of decomposition. With this kind of datasets, ELMOP is worse than NNOP for MZE and MAE, although its computational time is the lowest in binary decompositions. The best option from binary decompositions is SVMOP.When analysing threshold methods, our experiments show that the performance of

TABLE 8
Average ranking of the total computational time for each method.

Method	Average computational time			
	R_{D5}	R_{D10}	R_{D}	R_{OR}
SCV1V1	8.0833	8.0833	8.0833	6.1765
SVC1VA	6.8333	6.8333	6.8333	7.7647
SVR	12.0833	11.7500	11.9167	11.8235
CSSVC	6.3333	6.1667	6.2500	8.1765
SVMOP	9.5000	10.0000	9.7500	9.5294
NNOP	14.8333	9.3333	12.0833	12.4118
ELMOP	*2.8333*	*2.7500*	*2.7917*	*3.0588*
POM	**1.0000**	**1.2500**	**1.1250**	**1.7059**
NNPOM	15.9167	15.9167	15.9167	15.3529
SVOREX	4.0000	4.5000	4.2500	6.6471
SVORIM	4.9167	6.7500	5.8333	6.7059
SVORLin	6.6667	9.0000	7.8333	5.6471
KDLOR	12.5000	12.9167	12.7083	11.1765
GPOR	13.9167	12.8333	13.3750	13.5882
REDSVM	8.0000	9.8333	8.9167	9.2353
ORBALL	8.5833	8.0833	8.3333	7.0000

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

TABLE 9
Wilcoxon tests over discretised regression datasets ($Q = 5$ and $Q = 10$).

MZE				MAE				Time			
Method	w	d	l	Method	w	d	l	Method	w	d	l
GPOR	211	145	4	**REDSVM**	198	161	1	**POM**	357	0	3
SVOREX	149	205	6	SVORIM	195	162	3	**ELMOP**	314	11	35
SVORIM	137	208	15	GPOR	169	164	27	SVOREX	265	19	76
REDSVM	135	213	12	SVORLin	159	137	64	SVORIM	218	38	104
POM	118	169	73	SVOREX	156	177	27	**SVC1VA**	180	75	105
SVORLin	100	185	75	POM	152	136	72	SVORLin	171	55	134
SVMOP	76	219	65	**SVR**	146	165	49	CSSVC	169	95	96
SCV1V1	66	219	75	ORBALL	130	165	65	SCV1V1	156	52	152
KDLOR	65	225	70	**SVMOP**	96	154	110	ORBALL	152	70	138
ORBALL	65	193	102	KDLOR	75	186	99	REDSVM	134	66	160
SVR	59	213	88	SCV1V1	55	145	160	SVMOP	116	77	167
NNOP	35	195	130	ELMOP	54	135	171	NNOP	93	2	265
CSSVC	32	192	136	NNOP	52	159	149	SVR	79	42	239
SVC1VA	24	201	135	CSSVC	19	110	231	KDLOR	63	54	243
ELMOP	23	176	161	SVC1VA	17	106	237	GPOR	44	76	240
NNPOM	20	172	168	NNPOM	16	120	224	NNPOM	2	2	356

Best method of each family in the taxonomy is highlighted in bold face

GPOR is much lower in both MZE and MAE. SVM based threshold models are the best performing ones for both measures, SVOREX achieving the best results in MZE and very close to the best performing method, REDSVM, in MAE. In this case, the gap of performance between SVORIM and SVORLin is higher. Discarding POM and ELMOP, the lowest computationally time is associated to SVORLin. For ordinal regression datasets, the results of ORBALL are better with respect to the rest of methods than when consider the discretised datasets. With respect to REDSVM, its computational cost is high when compared to SVOR, ORBoost and SVC methods.

4.4 Discussion

Several ordinal regression methods (see Tables 9 and 10) can be emphasised according to their error (GPOR, SVOREX, SVORIM, REDSVM and SVMOP) or their computational time (POM, SVORLin or ELMOP). However, there are many factors that can influence the choice of the method, and all of them should be considered.

TABLE 10
Wilcoxon tests over ordinal regression datasets.

MZE				MAE				Time			
Method	w	d	l	Method	w	d	l	Method	w	d	l
SVOREX	86	159	10	**REDSVM**	88	160	7	**POM**	241	6	8
SVORIM	79	162	14	SVOREX	88	159	8	**ELMOP**	219	2	34
REDSVM	76	166	13	SVORIM	85	163	7	**SVORLin**	171	11	73
SCV1V1	75	170	10	ORBALL	74	145	36	SVOREX	154	10	91
SVMOP	74	166	15	**SVMOP**	68	176	11	SCV1V1	153	21	81
ORBALL	63	163	29	**SCV1V1**	64	169	22	SVORIM	153	11	91
GPOR	59	114	82	SVR	62	163	30	ORBALL	148	9	98
SVR	51	166	38	GPOR	55	126	74	SVC1VA	125	30	100
CSSVC	50	167	38	KDLOR	50	108	97	CSSVC	114	47	94
SVC1VA	47	164	44	NNOP	45	159	51	REDSVM	104	20	131
NNOP	43	160	52	CSSVC	42	162	51	SVMOP	97	17	141
KDLOR	39	120	96	SVC1VA	41	162	52	KDLOR	72	14	169
SVORLin	34	155	66	SVORLin	40	152	63	SVR	66	7	182
ELMOP	23	145	87	ELMOP	22	134	99	NNOP	57	9	189
NNPOM	22	131	102	POM	21	94	140	GPOR	34	28	193
POM	13	104	138	NNPOM	19	120	116	NNPOM	11	0	244

Best method of each family in the taxonomy is highlighted in bold face

First of all, POM is a linear model and, as such, it is very fast to train (with no associated hyperparameters), but its performance is significantly low (except for MZE in discretised regression datasets). This fact is important, given that, excluding the machine learning area, the POM and its variants are the most widely used ordinal regression methods [25], [81], [84], [88].

When dealing with large datasets, we conclude that POM is a good option, given the low computational cost needed. The results achieved for MZE and MAE are worse than those of other alternatives, but they are good enough when computational time is a priority. SVORLin can also be a good option, although a narrower range of cross-validation for C should be selected. Neural networks (NNPOM and NNOP) are generally beaten by their SVM counterparts, both in MZE and MAE. Moreover, the training time for these methods and GPOR is generally the highest.

Our study shows that the naïve approaches can obtain competitive performance and be difficult to beat for some datasets. SVC1V1 achieves very good MZE results for real ordinal regression datasets. However, SVM threshold models improves MAE and MZE results, as well as being simpler models. Indeed, all threshold models allow the visualisation of predicted projections together with the thresholds. This can be used for various purposes, from ranking patterns to trying to discover uncertain predictions (projections very close to class thresholds). This kind of analysis is generally more difficult with nominal models, such as SVC1V1. In general, the SVC1VA alternative has been shown to achieve worse results than SVC1V1 for the three measures evaluated (as previously shown in other studies [59]). CSSVC results are a bit better than those of SVC1VA, but still far from SVC1V1.

Binary decomposition approaches are shown to be good alternatives, especially SVMOP. However, as discussed in Subsection 3.2, their theoretical analysis is more difficult, and it is necessary to decide how to

combine different binary predictions.

Of all the threshold models analysed, SVOREX and SVORIM are the best. The computational time required by SVOREX is slightly lower, and it always achieves better results than SVORIM, except for MAE in discretised regression datasets. ORBALL shows a worse performance than SVOR methods. REDSVM is very competitive, but with a higher computational cost.

When comparing discretised regression datasets and real ordinal regression ones, some performance differences can be highlighted. For example, GPOR performance is seriously affected when dealing with real ordinal classification datasets. In general, SVM and ORBALL methods are more robust in the derived problems that can appear with these datasets. This is an important point, because many of the ordinal regression works in the literature make use of discretised regression sets, hiding some possible difficulties of the methods when dealing with problems such as imbalanced distributions.

When real ordinal regression datasets are considered, POM and GPOR performances decrease (both in MZE and MAE) drastically. Both models have one feature in common. They assume that their perturbation terms follow certain distribution functions. These distributional assumptions perform correctly in discretised regression datasets, but not for real ordinal regression datasets.

5 Conclusions

This paper offers an exhaustive survey of the ordinal regression methods proposed in the literature. The problem setting has been clearly established and differentiated from other ranking topics. After this, a taxonomy of ordinal regression methods is proposed, dividing them into three main groups: naïve approaches, binary decompositions and threshold models. Furthermore, the most important methods of each family (a total of 16 methods) are empirically evaluated in two kinds of datasets, 24 discretised regression datasets and 17 real ordinal regression ones.

The taxonomy proposed can help the researcher or the practitioner choose the best method for a concrete problem, considering also the empirical results herein provided. It can also assist researchers in developing and proposing new methods, providing a way to classify them and to select the most similar ones. The results presented in this paper confirm that there is no single method which performs the best in all possible datasets and problem requirements. However, these results can be used to discard some of the methods, especially those clearly presenting worse performance or too high computational time. We would like to stress certain methods: 1) SVC1V1 as representative of the naïve approaches, achieving an especially good MZE because of the recursive partitioning of all pairs of classes; 2) SVMOP achieves the best results from ordinal binary decomposition methods; 3) ELMOP or POM are a good option if the computational cost is a priority; and 4) SVOREX and SVORIM can be considered the best threshold models, showing competitive accuracy, MAE and time values. Finally, there is a website (http://www.uco.es/grupos/ayrna/orreview) collecting the implementations of the methods in this survey, the detailed results, the datasets and the corresponding statistical analysis.

References

[1] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.

[2] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., ser. Data Management Systems. Morgan Kaufmann (Elsevier), 2005.

[3] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*. Wiley-Interscience, 2007.

[4] J. A. Anderson, "Regression and ordered categorical variables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 46, no. 1, pp. 1–30, 1984.

[5] R. Bender and U. Grouven, "Ordinal logistic regression in medical research," *J. R. Coll. Physicians Lond.*, vol. 31, no. 5, pp. 546–551, 1997.

[6] ——, "Using binary logistic regression models for ordinal data with non-proportional odds," *J. Clin. Epidemiol.*, vol. 51, no. 10, pp. 809–816, 1998.

[7] W. M. Jang, S. J. Eun, C.-E. Lee, and Y. Kim, "Effect of repeated public releases on cesarean section rates," *J. Prev. Med. Pub. Health*, vol. 44, no. 1, pp. 2–8, 2011.

[8] O. M. Doyle, E. Westman, A. F. Marquand, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, S. Lovestone, S. C. Williams *et al.*, "Predicting progression of alzheimer's disease using ordinal regression," *PloS one*, vol. 9, no. 8, p. e105542, 2014.

[9] M. Pérez-Ortiz, P. A. Gutiérrez, C. García-Alonso, L. Salvador-Carulla, J. A. Salinas-Pérez, and C. Hervás-Martínez, "Ordinal classification of depression spatial hot-spots of prevalence," in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA)*, nov. 2011, pp. 1170 –1175.

[10] J. S. Cardoso, J. F. P. da Costa, and M. Cardoso, "Modelling ordinal relations with SVMs: an application to objective aesthetic evaluation of breast cancer conservative treatment," *Neural Networks*, vol. 18, no. 5-6, pp. 808–817, 2005.

[11] M. Pérez-Ortiz, M. Cruz-Ramírez, M. Ayllón-Terán, N. Heaton, R. Ciria, and C. Hervás-Martínez, "An organ allocation system for liver transplantation based on ordinal regression," *Applied Soft Computing Journal*, vol. 14, pp. 88–98, 2014.

[12] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, june 2011, pp. 585 –592.

[13] J. W. Yoon, S. J. Roberts, M. Dyson, and J. Q. Gan, "Bayesian inference for an adaptive ordered probit model: An application to brain computer interfacing," *Neural Networks*, vol. 24, no. 7, pp. 726–734, 2011.

[14] Y. Kwon, I. Han, and K. Lee, "Ordinal pairwise partitioning (opp) approach to neural networks training in bond rating," *Intelligent Systems in Accounting Finance and Management*, vol. 6, no. 1, pp. 23–40, 1997.

[15] K.-j. Kim and H. Ahn, "A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach," *Computers and Operations Research*, vol. 39, no. 8, pp. 1800–1811, 2012.

[16] H. Dikkers and L. Rothkrantz, "Support vector machines in ordinal classification: An application to corporate credit scoring," *Neural Network World*, vol. 15, no. 6, pp. 491–507, 2005.

[17] F. Fernández-Navarro, P. Campoy-Muñoz, M.-D. La Paz-Marín, C. Hervás-Martínez, and X. Yao, "Addressing the EU sovereign ratings using an ordinal regression approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2228–2240, 2013.

[18] M. J. Mathieson, "Ordinal models for neural networks," in *Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, ser. Neural Networks in Financial Engineering, J. M. A.-P. N. Refenes, Y. Abu-Mostafa and A. Weigend, Eds. World Scientific, 1996, pp. 523–536.

[19] M. Kim and V. Pavlovic, "Structured output ordinal regression for dynamic facial emotion intensity prediction," in *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010), Part III*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6313. Springer Berlin Heidelberg, 2010, pp. 649–662.

[20] O. Rudovic, V. Pavlovic, and M. Pantic, "Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2634–2641.

[21] ——, "Kernel conditional ordinal random fields for temporal segmentation of facial action units," in *Proceedings of the European Conference on Computer Vision Computer Vision (ECCV 2012). Part II*, ser. Lecture Notes in Computer Science, A. Fusiello, V. Murino, and R. Cucchiara, Eds., vol. 7584. Springer Berlin Heidelberg, 2012, pp. 260–269.

[22] H. Yan, "Cost-sensitive ordinal regression for fully automatic facial beauty assessment," *Neurocomputing*, vol. 129, pp. 334–342, 2014.

[23] Q. Tian, S. Chen, and X. Tan, "Comparative study among three strategies of incorporating spatial structures to ordinal image regression," *Neurocomputing*, vol. 136, no. 0, pp. 152 – 161, 2014.

[24] P. A. Gutiérrez, S. Salcedo-Sanz, C. Hervás-Martínez, L. Carro-Calvo, J. Sánchez-Monedero, and L. Prieto, "Ordinal and nominal classification of wind speed from synoptic pressure patterns," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 3, pp. 1008–1015, 2013.

[25] A. S. Fullerton and J. Xu, "The proportional odds with partial proportionality constraints model for ordinal response variables," *Soc. Sci. Res.*, vol. 41, no. 1, pp. 182–198, 2012.

[26] S. Baccianella, A. Esuli, and F. Sebastiani, "Feature selection for ordinal text classification," *Neural Comput.*, vol. 26, no. 3, pp. 557–591, 2014.

[27] C.-W. Seah, I. W. Tsang, and Y.-S. Ong, "Transductive ordinal regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1074–1086, 2012.

[28] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 115–132.

[29] W. Chu and S. S. Keerthi, "Support Vector Ordinal Regression," *Neural Comput.*, vol. 19, no. 3, pp. 792–815, 2007.

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

[31] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, and W.-B. Li, "Kernel discriminant learning for ordinal regression," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 6, pp. 906–910, 2010.

[32] J. C. Hühn and E. Hüllermeier, "Is an ordinal class structure useful in classifier learning?" *International Journal of Data Mining, Modelling and Management*, vol. 1, no. 1, pp. 45–67, 2008.

[33] A. Ben-David, L. Sterling, and T. Tran, "Adding monotonicity to learning algorithms may impair their accuracy," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6627–6634, 2009.

[34] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[35] P. A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez, "An experimental study of different ordinal regression methods and measures," in *7th International Conference on Hybrid Artificial Intelligence Systems*, 2012, pp. 296–307.

[36] S. Rajaram, A. Garg, X. S. Zhou, and T. S. Huang, "Classification approach towards ranking and sorting problems," in *Proc. of the 14th European Conference on Machine Learning (ECML)*, ser. Lecture Notes in Computer Science, vol. 2837, 2003, pp. 301–312.

[37] S. Rajaram and S. Agarwal, "Generalization bounds for k-partite ranking," in *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS2005)*, 2005, pp. 28–23.

[38] J. Fürnkranz, E. Hüllermeier, and S. Vanderlooy, "Binary decomposition methods for multipartite ranking," in *Proc. of the European Conference on Machine Learning (ECML)*, ser. Lecture Notes in Computer Science, vol. 578, no. I, 2009, pp. 359–374.

[39] K. Uematsu and Y. Lee, "Statistical optimality in multipartite ranking and ordinal regression," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. In Press, 2015.

[40] T.-Y. Liu, *Learning to Rank for Information Retrieval*. Springer-Verlag, 2011.

[41] H.-T. Lin and L. Li, "Reduction from cost-sensitive ordinal ranking to weighted binary classification," *Neural Comput.*, vol. 24, no. 5, pp. 1329–1367, 2012.

[42] Q. Hu, W. Pan, L. Zhang, D. Zhang, Y. Song, M. Guo, and D. Yu, "Feature selection for monotonic classification," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 69–81, 2012.

[43] Q. Hu, X. Che, L. Zhang, D. Zhang, M. Guo, and D. Yu, "Rank entropy-based decision trees for monotonic classification," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2052 –2064, 2012.

[44] W. Kotłowski, K. Dembczyński, S. Greco, and R. Słowiński, "Stochastic dominance-based rough set model for ordinal classification," *Inf. Sciences*, vol. 178, no. 21, pp. 4019–4037, 2008.

[45] W. Kotlowski and R. Slowiński, "On nonparametric ordinal classification with monotonicity constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2576–2589, 2013.

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

[47] C.-W. Seah, I. Tsang, and Y.-S. Ong, "Transfer ordinal label learning," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1863–1876, 2013.

[48] J. Alonso, J. J. Coz, J. Díez, O. Luaces, and A. Bahamonde, "Learning to predict one or more ranks in ordinal regression tasks," in *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML), Part I*, ser. Lecture Notes in Computer Science, W. Daelemans, B. Goethals, and K. Morik, Eds., vol. 5211. Springer Berlin Heidelberg, 2008, pp. 39–54.

[49] J. Verwaeren, W. Waegeman, and B. De Baets, "Learning partial ordinal class memberships with kernel-based proportional odds models," *Computational Statistics & Data Analysis*, vol. 56, no. 4, pp. 928–942, 2012.

[50] D. Devlaminck, W. Waegeman, B. Bauwens, B. Wyns, P. Santens, and G. Otte, "From circular ordinal regression to multilabel classification," in *Proceedings of the 2010 Workshop on Preference Learning (European Conference on Machine Learning, ECML)*, 2010.

[51] V. Torra, J. Domingo-Ferrer, J. M. Mateo-Sanz, and M. Ng, "Regression for ordinal variables without underlying continuous variables," *Information Sciences*, vol. 176, no. 4, pp. 465–474, 2006.

[52] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[53] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, August 2007.

[54] S. Kramer, G. Widmer, B. Pfahringer, and M. D. Groeve, "Prediction of ordinal classes using regression trees," *Fundamenta Informaticae*, vol. 47, no. 1-2, pp. 1–13, 2001.

[55] E. F. Harrington, "Online ranking/collaborative filtering using the perceptron algorithm," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML2003)*, 2003.

[56] J. Sánchez-Monedero, P. A. Gutiérrez, P. Tino, and C. Hervás-Martínez, "Exploitation of pairwise class distances for ordinal classification." *Neural Comput.*, vol. 25, no. 9, pp. 2450–2485, 2013.

[57] A. Agresti, *Analysis of ordinal categorical data*, ser. Wiley Series in Probability and Statistics. Wiley, 2010.

[58] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications*, vol. 16, pp. 264–280, 1971.

[59] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.

[60] H.-H. Tu and H.-T. Lin, "One-sided support vector regression for multiclass cost-sensitive classification," in *Proceedings of the Twenty-Seventh International Conference on Machine learning (ICML2010)*, 2010, pp. 49–56.

[61] S. B. Kotsiantis and P. E. Pintelas, "A cost sensitive technique for ordinal classification problems," in *Methods and applications of artificial intelligence (Proc. of the 3rd Hellenic Conference on Artificial Intelligence, SETN)*, ser. Lecture Notes in Artificial Intelligence, vol. 3025, 2004, pp. 220–229.

[62] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *J. of Machine Learning Research*, vol. 1, pp. 113–141, Sep. 2001.

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

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

[65] W. Waegeman and L. Boullart, "An ensemble of weighted support vector machines for ordinal regression," *International Journal of Computer Systems Science and Engineering*, vol. 3, no. 1, pp. 47–51, 2009.

[66] H. Wu, H. Lu, and S. Ma, "A practical SVM-based algorithm for ordinal regression in image retrieval," in *Proceedings of the eleventh ACM international conference on Multimedia (Multimedia2003)*, 2003, pp. 612–621.

[67] M. Pérez-Ortiz, P. A. Gutiérrez, and C. Hervás-Martínez, "Projection based ensemble learning for ordinal regression," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 681–694, 2014.

[68] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer, August 2001.

[69] M. Costa, "Probabilistic interpretation of feedforward network outputs, with relationships to statistical prediction of ordinal quantities," *Int. J. Neural Syst.*, vol. 7, no. 5, pp. 627–638, 1996.

[70] K. Crammer and Y. Singer, "Pranking with ranking," in *Advances in Neural Information Processing Systems*, vol. 14. MIT Press, 2001, pp. 641–647.

[71] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN2008, IEEE World Congress on Computational Intelligence)*. IEEE Press, 2008, pp. 1279–1284.

[72] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 2, pp. 513–529, 2012.

[73] W.-Y. Deng, Q.-H. Zheng, S. Lian, L. Chen, and X. Wang, "Ordinal extreme learning machine," *Neurocomputing*, vol. 74, no. 1–3, pp. 447–456, 2010.

[74] J. Sánchez-Monedero, P. A. Gutiérrez, and C. Hervás-Martínez, "Evolutionary ordinal extreme learning machine," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8073 LNAI, pp. 500–509, 2013.

[75] F. Fernandez-Navarro, A. Riccardi, and S. Carloni, "Ordinal neural networks without iterative tuning," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2075–2085, 2014.

[76] J. F. P. da Costa and J. Cardoso, "Classification of ordinal data using neural networks," in *Proceedings of the 16th European Conference on Machine Learning (ECML 2005)*, ser. Lecture Notes in Computer Science, J. Gama, R. Camacho, P. Brazdil, A. Jorge, and L. Torgo, Eds., vol. 3720. Springer Berlin Heidelberg, 2005, pp. 690–697.

[77] 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, January 2008.

[78] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[79] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Multi-task learning via conic programming," in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 737–744.

[80] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 42, no. 2, pp. 109–142, 1980.

[81] A. Agresti, *Categorical Data Analysis*, 2nd ed. John Wiley and Sons, 2002.

[82] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed., ser. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 1989.

[83] M. J. Mathieson, "Ordered classes and incomplete examples in classification," in *Proceedings of the 1996 Conference on Neural Information Processing Systems (NIPS)*, ser. Advances in Neural Information Processing Systems, T. P. Michael C. Mozer, Michael I. Jordan, Ed., vol. 9, 1999, pp. 550–556.

[84] R. Williams, "Generalized ordered logit/partial proportional odds models for ordinal dependent variables," *Stata Journal*, vol. 6, no. 1, pp. 58–82, March 2006.

[85] B. Peterson and J. Harrell, Frank E., "Partial proportional odds models for ordinal response variables," *Journal of the Royal Statistical Society*, vol. 39, no. 2, pp. 205–217, 1990, series C.

[86] G. Tutz, "Generalized semiparametrically structured ordinal models," *Biometrics*, vol. 59, no. 2, pp. 263–273, 2003.

[87] M. Dorado-Moreno, P. A. Gutiérrez, and C. Hervás-Martínez, "Ordinal classification using hybrid artificial neural networks with projection and kernel basis functions," in *7th International Conference on Hybrid Artificial Intelligence Systems (HAIS2012)*, 2012, p. 319–330.

[88] T. Van Gestel, B. Baesens, P. Van Dijcke, J. Garcia, J. Suykens, and J. Vanthienen, "A process model to develop an internal rating system: Sovereign credit ratings," *Decision Support Systems*, vol. 42, no. 2, pp. 1131–1151, 2006.

[89] J. D. Rennie and N. Srebro, "Loss functions for preference levels: Regression with discrete ordered labels," in *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*. Kluwer Norwell, MA, 2005, pp. 180–186.

[90] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," in *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN 99)*, vol. 1, 1999, pp. 97–102.

[91] A. Shashua and A. Levin, "Ranking with large margin principle: two approaches," in *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS2003)*, ser. Advances in Neural Information Processing Systems, no. 16. MIT Press, 2003, pp. 937–944.

[92] W. Chu and S. S. Keerthi, "New approaches to support vector ordinal regression," in *In ICML'05: Proceedings of the 22nd international conference on Machine Learning*, 2005, pp. 145–152.

[93] S. Agarwal, "Generalization bounds for some ordinal regression algorithms," in *Algorithmic Learning Theory*, ser. Lecture Notes in Artificial Intelligence (Lecture Notes in Computer Science). Springer-Verlag Berlin Heidelberg, 2008, vol. 5254, pp. 7–21.

[94] E. Carrizosa and B. Martin-Barragan, "Maximizing upgrading and downgrading margins for ordinal regression," *Mathematical Methods of Operations Research*, vol. 74, no. 3, pp. 381–407, 2011.

[95] B. Zhao, F. Wang, and C. Zhang, "Block-quantized support vector ordinal regression," *IEEE Trans. Neural Netw.*, vol. 20, no. 5, pp. 882–890, 2009.

[96] B. Gu, J.-D. Wang, and T. Li, "Ordinal-class core vector machine," *J. of Comp. Science and Technology*, vol. 25, no. 4, pp. 699–708, 2010.

[97] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *J. of Machine Learning Research*, vol. 6, pp. 363–392, 2005.

[98] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. In Press, 2015.

[99] J. S. Cardoso, R. Sousa, and I. Domingues, "Ordinal data classification using kernel discriminant analysis: A comparison of three approaches," in *11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 1, 2012, pp. 473–477.

[100] B.-Y. Sun, H.-L. Wang, W.-B. Li, H.-J. Wang, J. Li, and Z.-Q. Du, "Constructing and combining orthogonal projection vectors for ordinal regression," *Neural Processing Letters*, pp. 1–17, 2014.

[101] Y. Liu, Y. Liu, S. Zhong, and K. C. Chan, "Semi-supervised manifold ordinal regression for image ranking," in *Proceedings of the 19th ACM international conference on Multimedia (ACM MM2011)*. New York, NY, USA: ACM, 2011, pp. 1393–1396.

[102] Y. Liu, Y. Liu, and K. C. C. Chan, "Ordinal regression via manifold learning," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, W. Burgard and D. Roth, Eds. AAAI Press, 2011, pp. 398–403.

[103] Y. Liu, Y. Liu, K. C. C. Chan, and J. Zhang, "Neighborhood preserving ordinal regression," in *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service (ICIMCS12)*. New York, NY, USA: ACM, 2012, pp. 119–122.

[104] K. Crammer and Y. Singer, "Online ranking by projecting," *Neural Comput.*, vol. 17, no. 1, pp. 145–175, 2005.

[105] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Advances in Neural Information Processing Systems*, no. 19, 2007, pp. 865–872.

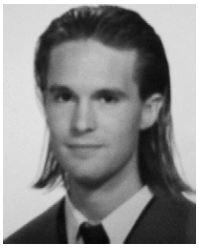
[106] H.-T. Lin and L. Li, "Combining ordinal preferences by boosting," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2009, pp. 69–83.

[107] ——, "Large-margin thresholded ensembles for ordinal regression: Theory and practice," in *Proc. of the 17th Algorithmic Learning Theory International Conference*, ser. Lecture Notes in Artificial Intelligence (LNAI), J. L. Balcázar, P. M. Long, and F. Stephan, Eds., vol. 4264. Springer-Verlag, October 2006, pp. 319–333.
[108] F. Xia, L. Zhou, Y. Yang, and W. Zhang, "Ordinal regression as multiclass classification," *International Journal of Intelligent Control and Systems*, vol. 12, no. 3, pp. 230–236, 2007.
[109] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. of Machine Learning Research*, vol. 4, pp. 933–969, 2003.
[110] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
[111] K. Dembczyński, W. Kotłowski, and R. Słowiński, "Ordinal classification with decision rules," in *Mining Complex Data*, ser. Lecture Notes in Computer Science, vol. 4944. Warsaw, Poland: Springer, 2008, pp. 169–181.
[112] A. Riccardi, F. Fernandez-Navarro, and S. Carloni, "Cost-sensitive AdaBoost algorithm for ordinal regression based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1898–1909, 2014.
[113] F. Fernández-Navarro, P. A. Gutiérrez, C. Hervás-Martínez, and X. Yao, "Negative correlation ensemble learning for ordinal regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1836–1849, 2013.
[114] W. Chu and Z. Ghahramani, "Preference learning with gaussian processes," in *Proceedings of the 22nd international conference on Machine learning (ICML2005)*, 2005, pp. 137–144.
[115] X. Chang, Q. Zheng, and P. Lin, "Ordinal regression with sparse bayesian," in *Proceedings of the 5th International Conference on Intelligent Computing (ICIC2009)*, ser. Lecture Notes in Computer Science, vol. 5755, 2009, pp. 591–599.
[116] P. Srijith, S. Shevade, and S. Sundararajan, "A probabilistic least squares approach to ordinal regression," *Lecture Notes in Artificial Intelligence*, vol. 7691, pp. 683–694, 2012.
[117] ——, "Validation based sparse gaussian processes for ordinal regression," *Lecture Notes in Computer Science*, vol. 7664, no. 2, pp. 409–416, 2012.
[118] ——, "Semi-supervised gaussian process ordinal regression," *Lecture Notes in Artificial Intelligence*, vol. 8190, no. 3, pp. 144–159, 2013.
[119] J. F. Pinto da Costa, R. Sousa, and J. S. Cardoso, "An all-at-once unimodal SVM approach for ordinal classification," in *Proceedings of the Ninth International Conference on Machine Learning and Applications (ICMLA2010)*. IEEE Computer Society Press, 2010, pp. 59–64.
[120] J. Cardoso and R. Sousa, "Classification models with global constraints for ordinal data," in *Proceedings of the Ninth International Conference on Machine Learning and Applications (ICMLA2010)*, 2010, pp. 71 –77.
[121] R. Sousa and J. Cardoso, "Ensemble of decision trees with global constraints for ordinal classification," in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA2011)*, 2011, pp. 1164–1169.
[122] S. Fouad and P. Tiňo, "Adaptive metric learning vector quantization for ordinal classification," *Neural Comput.*, vol. 24, pp. 2825–2851, 2012.
[123] S. Fouad and P. Tino, "Prototype based modelling for ordinal classification," in *Proceedings of the 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL2012)*, ser. Lecture Notes in Computer Science, H. Yin, J. Costa, and G. Barreto, Eds., vol. 7435. Springer Berlin Heidelberg, 2012, pp. 208–215.
[124] K. Dembczyański1 and W. Kotlowski, "Decision rule-based algorithm for ordinal classification based on rank loss minimization," in *Proceedings of the 2009 Workshop on Preference Learning (European Conference on Machine Learning, ECML)*, 2009.
[125] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[126] PASCAL, "Pascal (Pattern Analysis, Statistical Modelling and Computational Learning) machine learning benchmarks repository," 2011. [Online]. Available: http://mldata.org/
[127] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, May 2011.
[128] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. of Machine Learning Research*, vol. 5, pp. 975—-1005, 2004.
[129] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," *Neurocomputing*, vol. 50, no. 6, pp. 105–123, 2003.
[130] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, 2014.
[131] 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 (ISDA'09)*, 2009, pp. 283–287.
[132] L. Prechelt, "PROBEN1: A set of neural network benchmark problems and benchmarking rules," Fakultät für Informatik (Universität Karlsruhe), Tech. Rep. 21/94, 1994.
[133] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

Pedro Antonio Gutiérrez was born in Córdoba, Spain. He received the B.S. degree in Computer Science from the University of Sevilla, Spain, in 2006, and the Ph.D. degree in Computer Science and Artificial Intelligence from the University of Granada, Spain, in 2009. He is currently an Assistant Professor in the Department of Computer Science and Numerical Analysis, University of Córdoba, Spain. His current research interests include patter recognition, evolutionary computation, and their applications.

María Pérez-Ortiz was born in Córdoba, Spain, in 1990. She received the B.S. degree in computer science in 2011 and the M.Sc. degree in intelligent systems in 2013 from the University of Córdoba, Spain, where she is currently pursuing the Ph.D. degree in computer science and artificial intelligence in the Department of Computer Science and Numerical Analysis. Her current interests include a wide range of topics concerning machine learning and pattern recognition.

Javier Sánchez-Monedero was born in Córdoba (Spain). He received the B.S in Computer Science from the University of Granada, Spain, in 2008 and the M.S. in Multimedia Systems from the University of Granada in 2009, where he obtained the Ph.D. degree on Information and Communication Technologies in 2013. He is working as researcher with the Department of Computer Science and Numerical Analysis at the University of Córdoba. His current research interests include computational intelligence and distributed systems.



Francisco Fernández-Navarro (M'13) was born in Córdoba, Spain, in 1984. He received the M.Sc. degree in computer science from the University of Córdoba, Spain, in 2008, the M.Sc. degree in artificial intelligence from the University of Málaga, Spain, in 2009 and the Ph.D. degree in computer science and artificial intelligence from the University of Málaga, in 2011. He is currently a Research Fellow in computational management with the European Space Agency, Noordwijk, The Netherlands. His current research interests include neural networks, ordinal regression, imbalanced classification and hybrid algorithms.

César Hervás-Martínez was born in Cuenca, Spain. He received the B.S. degree in Statistics and Operations Research from the "Universidad Complutense", Madrid, Spain, in 1978, and the Ph.D. degree in Mathematics from the University of Seville, Spain, in 1986. He is currently a Professor of Computer Science and Artificial Intelligence in the Department of Computer Science and Numerical Analysis, University of Córdoba, and an Associate Professor in the Department of Quantitative Methods, School of Economics, University of Córdoba. His current research interests include neural networks, evolutionary computation, and the modelling of natural systems.