Prog Artif Intell (2016) 5:171–179
DOI 10.1007/s13748-016-0088-y

REGULAR PAPER

Current prospects on ordinal and monotonic classification

Pedro Antonio Gutiérrez[1] · **Salvador García**[2]

Received: 8 February 2016 / Accepted: 21 February 2016 / Published online: 4 March 2016

Abstract Ordinal classification covers those classification tasks where the different labels show an ordering relation, which is related to the nature of the target variable. In addition, if a set of monotonicity constraints between independent and dependent variables has to be satisfied, then the problem is known as monotonic classification. Both issues are of great practical importance in machine learning. Ordinal classification has been widely studied in specialized literature, but monotonic classification has received relatively low attention. In this paper, we define and relate both tasks in a common framework, providing proper descriptions, characteristics, and a categorization of existing approaches in the state-of-the-art. Moreover, research challenges and open issues are discussed, with focus on frequent experimental behaviours and pitfalls, commonly used evaluation measures and the encouragement in devoting substantial research efforts in specific learning paradigms.

Keywords Machine learning · Ordinal classification · Ordinal regression · Monotonic classification · Evaluation measures

✉ Pedro Antonio Gutiérrez
pagutierrez@uco.es

Salvador García
salvagl@decsai.ugr.es

[1] Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein building, 14071 Córdoba, Spain

[2] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

1 Introduction

In the field of machine learning, classification problems are focused on assigning each input vector to one of a finite number of discrete categories [8], given a set of training data with pre-labelled examples. In this context, special considerations should be taken into account if the labels exhibit an ordering relation, i.e. they are naturally ordered according to the variable definition. For example, financial trading could be assisted by ordinal classification techniques predicting not only a binary decision of buying an asset, but also the amount of investment. The decision could categorised by {"no investment", "low investment", "medium investment", "huge investment"}. Machine learning methods should consider the natural order among the classes and penalise differently the errors. Confusing a "no investment" instance with a "huge investment" should be associated a higher cost than a "little investment" prediction for the same instance. Ordinal classification [34] (also known as ordinal regression) deals with this kind of problems by trying to exploit the ordinal relation between labels and imposing it in the models to learn.

The classification with monotonicity constraints, also known as monotonic classification [6], is an ordinal classification problem where a monotonic restriction can be found: a higher value of an attribute in an example, fixing other values, should not decrease its class assignment. The monotonicity of relations between the dependent and explanatory variables is very usual as a prior knowledge form in data classification [40]. To illustrate this, consider a credit card application [14]. A \$1000 to \$2000 income may be considered a medium value of income in a data set. If a customer A has a medium income, a customer B has a low income (i.e. less than \$1000) and the rest of input attributes remain the same, there is a relationship of partial order between A and B: $B \prec A$. Considering that

the application estimates the lending quantities as the output class, it is quite obvious that the loan that the system should give to customer B should not be greater than the one given to customer A. If it is, a monotonicity constraint is violated in the decision.

At least, two important reasons are identified for explaining why knowledge about monotonicity should be exploited in a learning task [7]. First, monotonicity imposes constraints on the prediction function. This decreases the size of the hypothesis space and also the complexity of the model. Second, in many cases, the domain experts decide the acceptance or rejection of the trained models based on their consistency with respect to the domain knowledge, regardless of their accuracy.

In this paper, we will give a quick snapshot of ordinal and monotonic classifications problems, emphasizing the issues they have in common, the evaluation measures and the most important approaches already proposed. Also, the open issues and present trends in both related classification problems will be examined, suggesting several open challenges and new directions to devote efforts in the near future.

The rest of the paper is organized as follows. We first formalize the ordinal and monotonic classification problems (Sect. 2). Then, in Sect. 3, we describe several performance metrics widely used in the two problems. Afterwards, we enumerate the main methods proposed for tackling these problems (Sect. 4). Open issues and challenges are pointed out in Sect. 5. Finally, some concluding remarks are given in Sect. 6.

2 Problem definition

A standard classification problem consists of predicting the category y of an input pattern $\mathbf{x}$, where $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$ and $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$. The objective is to find a classifier $r : \mathcal{X} \to \mathcal{Y}$ to categorise new patterns. The classifier has to be learnt from a training set of N points, $D = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$. For ordinal and monotonic classification problems, a natural label ordering is included in the form $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_Q$, where $\prec$ is an order relation. The position of the label in the ordinal scale is used by many ordinal classification measures and algorithms, which can be expressed by $\mathcal{O}(\mathcal{C}_q) = q, q = 1, \ldots, Q$.

The difference between ordinal classification and other supervised problems is now established. Comparing the problem to nominal classification, the order between class labels makes that two different elements of the set $\mathcal{Y}$ can be always compared by using the relation $\prec$, and this is not possible under the nominal classification setting. If compared to regression (where $y \in \mathbb{R}$), although the standard $<$ operator can be used to order real values in $\mathbb{R}$, labels in ordinal classification ($y \in \mathcal{Y}$) do not carry metric information, so it is not possible to establish the distance between two given labels (following the example of Sect. 1, the distance between "low investment" and "medium investment" could be significantly higher than that between "no investment" and "low investment", and there is no principle way to measure it without a priori knowledge of the problem).

The case of monotonic classification is a particular case of ordinal classification, where there are monotonicity constraints between features and decision classes, i.e. $\mathbf{x} \succeq \mathbf{x}' \rightarrow f(\mathbf{x}) \geq f(\mathbf{x}')$ [40], where $\mathbf{x} \succeq \mathbf{x}'$ means that $\mathbf{x}$ dominates $\mathbf{x}'$, i.e. $x_k \geq x'_k, k = 1, \ldots, K$.

In monotonic classification, we have to define the concepts of monotonic classifier and monotonic data set. A monotonic classifier is one that will not violate monotonicity constraints, those given previously. There will be pure monotonic classifiers, whose decisions will be always monotonic between the independent variables and the dependent one; and approximate monotonic classifiers, which try to learn models as monotonic as possible, namely, predictions with the lowest number of monotonic violations.

A training data set D is monotonic if and only if all the pairs of examples i, j are monotonic with respect to each other [5]: $\mathbf{x}_i \succeq \mathbf{x}_j \rightarrow y_i \geq y_j, \forall_{i,j}$. Some monotonic classifiers require pure monotonic data sets to successfully learn, although there are others that are capable of learning from non-monotonic data sets as well. Even using pure monotonic data sets as input, there are monotonic classifiers that build approximate monotonic models.

3 Performance metrics

Ordinal and monotonic classifiers can be evaluated using different metrics [3,12,18,48]. The two most common metrics are the mean zero-one error (MZE) and the mean absolute error (MAE). The first one is defined as:

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

where y_i and y_i^* are the true and predicted labels, respectively, and Acc is the accuracy of the classifier. The range of MZE is [0, 1]. It is related to global performance, but the order is not considered. It is also known as 0/1 loss or standard misclassification rate. A way to include order information in the evaluation is to make use of the MAE metric, which is the average deviation in absolute value of the predicted rank ($\mathcal{O}(y_i^*)$) from the true one ($\mathcal{O}(y_i)$) [3]:

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

where MAE $\in [0, Q-1]$. This measure is also referred to as absolute error or rank loss.

Many binary classifiers assign scores to the different examples, and then a threshold is used for separating negative samples from positive ones. In this context, the area under the receiver operating characteristics (AUC) curve is one of the most commonly used metrics for evaluating the performance of a binary classifier, independently of the threshold used. Basically, it estimates the probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one [23]. AUC has been extended to ordinal classification problems [61], by assuming classifiers based on a scoring function with $Q-1$ different thresholds (threshold models), in such a way that each class corresponds to an interval delimited by these thresholds (more details on threshold models are given in Sect. 4.1). Consequently, AUC for ordinal classification is based on measuring the probability that a pattern from a given class is correctly ranked (according to the score function) with respect to patterns of the remaining classes.

The same measures described above are also used in monotonic classification for estimating the generalization performance of the trained models over test data. This behaviour could produce some negative effects, which will be discussed in Sect. 5.2.

Regarding the quantification of the monotonicity in predictions, which is a particular condition in monotonic classification when noise is present, there are several metrics.

The first is the non-monotonic index (NMI). This measurement was defined by Ben-David in [5] as the rate of number of violations of monotonicity divided by the total number of pairs of examples (excluding the pairs formed by themselves) in a data set: $\mathrm{NMI}(D) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} m_{i,j}}{N^2-N}$, where $m_{i,j}$ is equal to 1 if the pair formed by $\mathbf{x_i}$ and $\mathbf{x_j}$ is non-monotonic. To aggregate standard estimation scores of decision trees with quantification of monotonicity, the *order-ambiguity-score* (A) is computed, as shown in the next equation, using the concept of NMI:

$$A = \begin{cases} 0, & \text{if NMI} = 0, \\ -(\log_2 \mathrm{NMI})^{-1}, & \text{otherwise.} \end{cases}$$

Other alternative definitions are the following ones:

- Non-monotonicity index 1 (NMIO) [19], defined as the number of clash-pairs divided by the total number of pairs of examples in the data set: $\mathrm{NMIO} = \frac{1}{N(N-1)}\sum_{\mathbf{x}\in D}\mathrm{NClash}(\mathbf{x})$. $\mathrm{NClash}(\mathbf{x})$ is the number of examples from D that do not meet the monotonicity restrictions (or clash) with respect to $\mathbf{x}$.
- Non-monotonicity index 2 (NMIT) [47], slightly different to the previous, is defined as the number of non-monotonic examples divided by the total number of examples: $\mathrm{NMIT} = \frac{1}{N}\sum_{\mathbf{x}\in D}\mathrm{Clash}(\mathbf{x})$, where $\mathrm{Clash}(\mathbf{x}) = 1$ if $\mathbf{x}$ clashes with at least one example in D, and 0 otherwise. If $\mathrm{Clash}(\mathbf{x}) = 1$, $\mathbf{x}$ is called a non-monotone example.

We stress that these three last indices range in the unit interval, so they can be conveniently expressed as percentages.

4 Main methods

In this section, we briefly describe the most relevant methods proposed in the specialized literature for ordinal and monotonic classification.

4.1 Ordinal classification classifiers

According to [34], ordinal classification methods can be classified into the following families:

- *Naïve approaches* include those methods which simplify ordinal classification into other standard problems, by making some assumptions. For example, all the different labels $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$ can be mapped to real values $\{r_1, r_2, \ldots, r_Q\}$ [58], where $r_i \in \mathbb{R}$, and then standard regression techniques [8] (such as neural networks, support vector regression...) can be applied. Another option is to consider nominal classification and simply ignore ordering information. Moreover, different misclassification costs can be assigned according to order information, resulting in cost-sensitive classification, where the cost matrix is usually related to the absolute difference of ranks of true and predicted classes (in a similar way to the costs assumed by MAE). The three options imply different assumptions which may hamper the learning process: regression and cost-sensitive classification methods assume a distance or a cost between labels which is generally unknown (being more sensitive to the representation of the labels rather than its ordering [35]), nominal classification ignores order information (generally requiring more training data [35]).
- *Ordinal binary decompositions* are based on decomposing the ordinal classification task into a set of binary classification subtasks, in the same vein that multiclass classification problems are frequently simplified into several binary tasks using the well-known *One-Versus-One* or *One-Versus-All* schemes [29]. Ordering information can be imbued in these decompositions by considering, for example, that a pattern of class $\mathcal{C}_q$ should be classified as positive by all binary classifiers corresponding to classes with a lower or equal rank, i.e. $f_k(\mathbf{x}) = 1, \forall_k \mathcal{C}_k \preceq \mathcal{C}_q$ (*Ordered Partitions* scheme) [34].

- *Threshold models* are based on assuming that the ordinal class labels originate from consecutive intervals of an unobservable one-dimensional latent variable. These models learn two different elements from training data: (1) a one-dimensional projection function corresponding to an estimation of the latent variable, and (2) a set of $Q-1$ thresholds which divides the projection in Q classes (each class being defined by an interval). This scheme has been considered by many proposals in the literature, adapting linear logistic regression [46], support vector machines [16], discriminant analysis [57] or Gaussian processes [15] to the context of ordinal classification.
- *Augmented binary classification* the reduction framework of Lin and Li [43] approaches ordinal classification problem by reducing it to binary classification with additional input variables and specific weights for the extended patterns. A previous method exploiting the same idea is the data replication method of Cardoso et al. [11], the main difference being that it is limited to the absolute cost, whereas the framework of Lin and Li [43] can be used with any V-shaped cost matrix. On the other hand, data replication proposal includes a parameter s which limits the number of adjacent classes considered [11], in order to reduce the number of additional data points generated by the approach.

4.2 Monotonic classifiers

There are five main families of methods that deal with monotonicity constraints in classification:

- *Instance-based learning* methods are pioneers and well-known in the field of monotonic classification. Next, we will describe the three most important techniques in detail: OLM, OSDL and k-NN.
  - The ordinal learning model (OLM) [4] is a very simple algorithm that learns ordinal concepts by eliminating non-monotonic pairwise inconsistencies. The generated concepts can be viewed as rules. During the learning phase, each example is checked against every rule in a rule-base, which is initially empty. If an example is inconsistent with a rule in the rule-base, one of them is selected at random while the other is discarded, but if the example is selected, it must be checked for consistency against all the other monotonicity rules. If it passes this consistency test, it is added as a rule. Consequently, the rule-base is kept monotonic at all times. Classification is done conservatively. All the rules are checked in decreasing order of class values against an attribute vector, and the vector is classified as the class of the first rule that covers it. If such a rule does not exist, the attribute vector is assigned the lowest possible class.
  - The ordinal stochastic dominance learner [42] (OSDL) is based on the concept of ordinal stochastic dominance. The stochastic order computes when a random variable is bigger than another. Considering this order, stochastic dominance can be established as a form of stochastic order. In this case, a probability distribution over possible predictions can be ranked. The ranking depends on the nature of the data set. Stochastic dominance refers to a set of relations that may hold between a pair of distributions.
  - The monotonic k-NN was proposed in [22]. This method consists of two steps. In the first step, the training data is made monotone by relabelling as few cases as possible. This relabelled data set may be considered as the monotone classifier with the smallest error index in the training data. In the second step, a modified nearest neighbour rule is used to predict the class labels of new data, so that violations of the restrictions of monotonicity will not occur.

  Recently, some approaches that hybridize rule induction and instance-based learning, such as the nested generalized example learning, have appeared in monotonic classification [30,31].
- *Decision trees* A monotone extension of ID3 (MID) was proposed by Ben-David [5], using an additional impurity measure for splitting the total ambiguity score. However, the resulting tree may not be monotone anymore even when starting from a monotone data set. MID defines the *total-ambiguity-score* as the sum of the entropy score of ID3 and the order-ambiguity-score. This last score is defined in terms of the NMI of the tree (see Sect. 3).
  Makino et al. [44] proposed a monotone (or positive) decision tree (P-DT) and a quasi-monotone (quasi-positive) decision tree (QP-DT) extension of ID3 in the two-class setting. They start from a monotone training set and demand, in the case of QP-DT, that monotonicity is (only) guaranteed on this training set, while in the case of P-DT the tree (or equivalently, the derived rule base) is required to be monotone. These methods have been nontrivially extended in [53] to the multi-class problem, accommodating also continuous attributes. In addition to the fact that these approaches start from a monotone training set, the main technique for guaranteeing (quasi-)monotonicity is by adding at each step, if necessary, new data generated from the data in the previous step.
  A splitting criteria thought for monotonic classification has been proposed in [10]. The criterion aims at reducing the numbers of non-monotone pairs of points in the resulting branches. It chooses the split with the least number of conflicts. Another way to achieve monotone classi-

fication models in a post-processing step is by pruning classification trees [25]. This method prunes the parent of the non-monotone leaf that provides the largest reduction in the number of non-monotonic leaves' pairs. Here, similar accuracy is reported, with increased comprehensibility. Isotonic regression is also used for relabelling non-monotone leaf nodes of the decision tree [38].

As for explicit monotonic trees, we can find some representatives proposed in the literature. MDT [41] aimed to predict the implicit ordering in terms of pair comparison in the original classification. In [36], the authors propose a rank generalization of Shannon mutual information, namely rank mutual information and underline that this measure is both sensitive to monotonicity and robust to noisy data. Then, this measure is used to build binary tree classifiers guaranteed to have a weak form of monotonicity (rule monotonicity), in the case the starting data set is monotone consistent. They call this algorithm REMT and show that it behaves well compared to both monotone and non-monotone classifiers. An extension of the interval valued attribute decision tree to deal with monotonic classification is given in [63], which selects extended attributes by minimizing rank mutual information to generate a decision tree. Recently, in [45], the authors presented a binary tree classifier, RDMT(H), parametrized by a discrimination measure H used for splitting and other three pre-pruning parameters. According to them, RDMT(H) guarantees a weak form of monotonicity on the resulting tree.

- *Ensemble learning* techniques have been proposed for classification with monotonicity constraints. For instance, a boosting-like technique for ordinal classification problems related to decision rules has been proposed in [21,39]. Ensembles of bagged decision rules have been considered in [9] and bagged decision trees in [56]. This last paper considered global constraints in ordinal classification by imposing the ordinal constraints in a decision function and avoiding over-regularised decision spaces. A straightforward scheme based on Random Forest and ensemble pruning was proposed in [33]. Finally, in [54], the authors developed a method of fusing monotonic decision trees.
- *Neural networks* have been also applied on monotonic classification. Total and partial monotonic neural networks were examined in [20], and an adaptation of neural networks that imposes monotonicity constraints on the weights connecting the hidden layer with the output layer was presented in [26].
- *Data preprocessing and construction* Another trend in monotonic classification is to preprocess the data [32] in order to "monotonize" the data set, rejecting or relabelling the examples that violate the monotonic restrictions. There are two consolidated techniques for obtaining monotonic data sets, either generating artificial data [47,52] or by relabelling existing data sets [22,24,55]. The second option is the preferred for addressing real data sets with classifiers that exclusively work with monotonic data.

5 Open issues

This section discusses some specific problems and issues associated to ordinal and monotonic classification, establishing aspects of the field which should receive further attention by the research community.

5.1 Open issues in ordinal classification

Focusing on ordinal classification, the main problems are related to the performance metrics and the data sets used for evaluating the classifiers. Specifically:

- First of all, it is important to outline the necessity of taking ordering information into account. Many works in the machine learning field use ordinal classification data sets, ignoring the order of the categories. This can decrease the performance of the obtained model [34]. Some authors have previously studied whether there are performance improvements when considering order information. In [37], ordinal meta-models were compared against nominal ones, concluding that such ordinal methods may yield better performance. Indeed, much more differences can be found when considering specific ordinal classification methods (instead of meta-models) [34]. Another study [7] argues that ordinal classifiers may not present meaningful advantages over the analogue non-ordinal methods, based on accuracy and Cohen's Kappa statistic [17]. However, the results in [34] show that statistically significant differences are found when using measures which take the order into account, such as the MAE. In this way, it is extremely important to consider ordinal performance metrics to evaluate the benefits of applying ordinal classifiers.
- Independently of the performance differences, there are additional advantages on the use of ordinal classification models. For example, threshold models allow projection of the patterns into a real line, according to the latent variable value estimated for each pattern. This additional information can be very useful to detect uncertain predictions (close to the thresholds of its category) or to rank patterns in the same class.
- Proportional odds model (POM) is a linear model based on extending binary logistic regression to ordinal classification [46]. As such, its training is very fast but its performance is generally low [34], because many real

world problems require nonlinear decision boundaries. This fact is important, given that the POM and its variants are the most widely used ordinal classification methods in areas such as medical sciences or psychology [27,60,62].

- Public repositories (such as UCI [2], Keel [1], or `mldata.org` [50]) include benchmark classification data sets with ordinal classification problems. Indeed, there are many previous works where these data sets are treated as standard classification. A careful examination of the data sets in these repositories is needed to understand the nature of the target variable and separate ordinal classification tasks from standard multiclass classification (although for monotonic classification it makes sense to consider binary problems, ordinal classification problems must be, at least, three-class problems). However, there are some ordinal classification works [15,16,43] which consider the repository provided by Chu et al. [15]. These data sets 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 or equal width. Validating new algorithms using only these data sets can result in misleading conclusions, because it is clear that they can be simpler than real ordinal classification problems. When using equal frequency binning, class imbalance is suppressed, given that all classes are assigned the same number of patterns. For equal width binning, all classes are assigned intervals of the same width in the latent variable, simplifying the problem. Furthermore, there are observed values of the actual target regression variable (although they are ignored), so the classification problem can be simpler than problems where these values are not available and there are only categories. In any case, we do not neglect the opportunity of using these data sets to check how the algorithms perform in a more controlled environment, but we think they should be complemented with real ordinal classification data sets in order to test the performance of the classifiers in realistic settings.
- Finally, the different performance metrics used to evaluate ordinal classifiers make different assumptions. For example, MAE assigns proportional costs to all pairs of consecutive categories, in such a way that misclassifying a pattern of class $\mathcal{C}_3$ as class $\mathcal{C}_1$ is exactly twice more costly than assigning it a $\mathcal{C}_2$. This may not be the case for many real problems, where the costs of misclassifications may be very different depending on the classes evaluated. One possibility could be the use of association metrics [18], which evaluate the relative order of the patterns, but not the exact labels, i.e. if patterns are well sorted using the ordinal classifier with respect to sorting established by the true labels. Again, association metrics should be used together with MAE or other alternative metrics, because they evaluate different aspects of the classifiers (e.g. a classifier which shifts all the predictions one class in the ordinal scale will score the same value for association metrics).
- Ordinal classification data sets are unbalanced in nature, because extreme classes tend to be associated to rare events. Uneven pattern distributions pose a serious hindrance for classifier training, in such a way that minority classes tend to be ignored by the obtained classifiers. There have been many efforts for tackling this problem in the binary or multiclass classification contexts [13,28], but few works adapts these techniques to ordinal classification [51]. Specific characteristics of ordinal classification tasks should be taken into account when developing new strategies for alleviating this problem.

5.2 Open issues in monotonic classification

This section will be devoted to point out some of the existing open issues we have detected in monotonic classification.

- As we have indicated before, most of the same measures described for standard ordinal classification are usually used in experimental comparisons of monotonic classifiers for estimating the generalization performance, such as MZE or MAE. This can cause a major deviation and misinterpretation of the results reported over real problems and learned concepts. It is worth mentioning that any performance metric is estimated over a set of test examples, and these test examples are obtained by using statistical validations applied over the original data. Thus, it seems obvious to expect that if the original data set has inconsistencies or non-monotonicity violations, part of these flaws will be inherited in the test partitions. It is frequent and avoidable in standard classification, but it is more critical in monotonic classification. The errors derived from misclassifications of test examples that are really non-monotonic will influence the final estimate of performance measure. If we want to learn a monotonic model, we hope to predict unseen examples satisfying monotonic constraints. In this manner, the experimental evaluations should achieve conclusions from a link between measures of generalization performance and measures for estimating the degree of monotonicity in the predictions, such as NMI.
- An attempt of reducing the previous effect was suggested in [30], when tackling data sets with noise and possible non-monotonic violations. The measures MAcc and MMAE (monotonic accuracy and MAE, respectively) were used to estimate the errors over test data. Both do not consider the non-monotonic comparable examples in the estimate. The reason for this is to ensure that future examples will fulfil the monotonicity assumption. These metrics serve as monotonicity level measurements of the

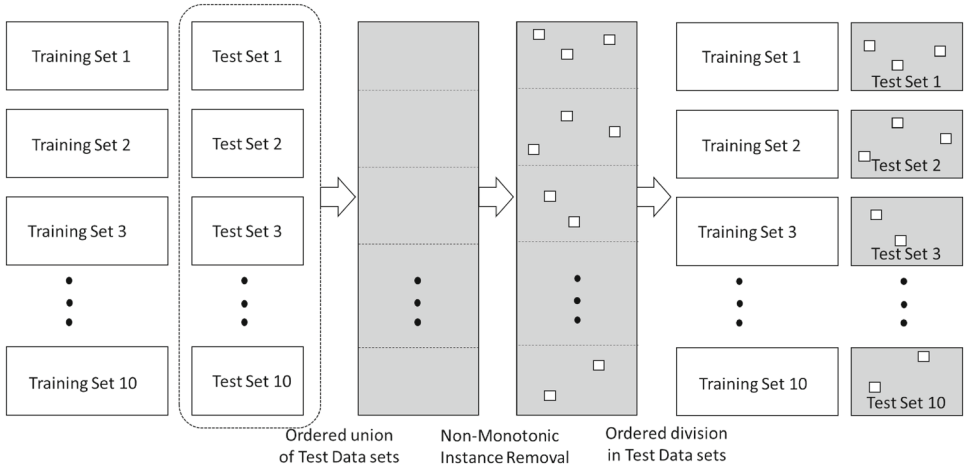

Fig. 1 Process of transformation of the 10-fcv to the new one by removing the non-monotonic instances from the test data sets

predictions carried out. To address this, the validation process followed is presented in Fig. 1 where the partitions used in 10-fcv are modified conserving the training data sets but removing the non-monotonic instances in the test data sets. The process of conflictive instance removal searches for fair comparisons and it is based on a deterministic greedy algorithm, avoiding randomness (see Algorithm 1).

Algorithm 1 Greedy Non-Monotonic Instances Removal Algorithm for test partitions.

```
function GreedyRemoval(D - data set)
  while NumberOfTotalCollisions(D)>0 do
    maxColis=0, x_selected=0;
    for each instance x_i in D do
      Colis=NumberOfCollisionsProduced(x_i,D);
      if Colis>maxColis then
        maxColis=Colis, x_selected=x_i;
      end if
    end for
    D = D - x_selected;
  end while
  return D
end function
```

- NMI is a well-known metric used in monotonic classification for estimating the degree of monotonicity in a set of predictions. It is also used to compute the degree of monotonicity in models, especially in interpretable models such as decision trees [5,36] or decision rules [4]. More complex models built from other algorithms such as ensembles or neural networks also contain hidden monotonicity violations in their models. This emphasizes the fact that obtaining monotonic predictions is as important as obtaining monotonic models.
- The most widely used scheme for preparing the data to learners that explicit require complete monotonic data is to relabel the data [24]. However, relabelling is not always the best approach to preprocess data that contains noise, inconsistencies and harmful examples [32]. The problem may not be only present in the class label, so other techniques such as edition, noise filtering and attribute values correction could work well in these cases.
- Possible extensions of the monotonic classification problem could consider different priorities among the explanatory attributes or different degrees of non-monotonicity among examples. The former extension refers to prioritize some attributes over others, depending on the background taken from the problem itself. As an example in credit risk, it may be more critical to predict a favourable loan to a first customer with lower income than another customer whose loan was denied, instead of predicting the same when considering the attribute "assets", keeping the remaining attribute values unchanged. The later extension is very related to the former one, because it assumes the existence of different degrees of non-monotonicity between two examples, which must be calculated by using the explanatory variables in one way or another. Also, the spacial separation of data points could influence this factor.
- Currently, monotonic classification is seen as a natural extension of classical or ordinal classification. Other predictive learning paradigms that require some interpretation of the results can benefit of monotonic models or predictions in certain real applications, such as Subgroup Discovery [49], Semi-Supervised learning [59] or Multi-Label learning.

6 Conclusions

This paper has presented a review and analysis of two supervised classification tasks highly related: ordinal and monotonic classification. Both are concerned with the classification of patterns into naturally ordered categories, although the latter considers constraints of monotonicity between input and target variables. A common framework

and notation is given for both kinds of problems, and the main existing techniques used for them are categorized and reviewed.

Moreover, the paper has uncovered some of the pitfalls and problems hidden in both fields, which are mainly related to the performance metrics, the data sets and the experimental design used for the evaluation of new ordinal or monotonic classification methodologies. We think that our analysis can serve as a motivation for developing specific experimental strategies for these tasks or as a set of recommendations for the application of existing ordinal and monotonic methodologies to other fields of study.

Acknowledgments This work is supported by the National Research Projects TIN2014-57251-P and TIN2014-54583-C2-1-R of the Spanish Ministry of Economy and Competitiveness (MINECO), by FEDER Funds and by the P11-TIC-7508 project of the Junta de Andalucía, Spain.

References

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Multiple-Valued Logic Soft Comput. **17**(2–3), 255–287 (2010)
2. Asuncion, A., Newman, D.: UCI Machine Learning Repository (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA'09), pp. 283–287 (2009)
4. Ben-David, A.: Automatic generation of symbolic multiattribute ordinal knowledge-based dsss: methodology and applications. Decision Sci. **23**, 1357–1372 (1992)
5. Ben-David, A.: Monotonicity maintenance in information-theoretic machine learning algorithms. Mach. Learn. **19**(1), 29–43 (1995)
6. Ben-David, A., Sterling, L., Pao, Y.H.: Learning, classification of monotonic ordinal concepts. Comput. Intell. **5**, 45–49 (1989)
7. Ben-David, A., Sterling, L., Tran, T.: Adding monotonicity to learning algorithms may impair their accuracy. Expert Syst. Appl. **36**(3), 6627–6634 (2009)
8. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
9. Blaszczynski, J., Slowinski, R., Stefanowski, J.: Ordinal classification with monotonicity constraints by variable consistency bagging. In: Proceedings of 7th International Conference on Rough Sets and Current Trends in Computing, RSCTC 2010, Warsaw, Poland, June 28–30, 2010, pp. 392–401 (2010)
10. Cao-Van, K., De Baets, B.: Growing decision trees in an ordinal setting. Int. J. Intell. Syst. **18**(7), 733–750 (2003)
11. Cardoso, J.S., da Costa, J.F.P.: Learning to classify ordinal data: the data replication method. J. Mach. Learn. Res. **8**, 1393–1429 (2007)
12. Cardoso, J.S., Sousa, R.G.: Measuring the performance of ordinal classification. Int. J. Pattern Recognit. Artif. Intell. **25**(8), 1173–1195 (2011)
13. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
14. Chen, C.C., Li, S.T.: Credit rating with a monotonicity-constrained support vector machine model. Expert Syst. Appl. **41**(16), 7235–7247 (2014)
15. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. J. Mach. Learn. Res. **6**, 1019–1041 (2005)
16. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Comput. **19**(3), 792–815 (2007)
17. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. **20**(1), 37–46 (1960)
18. Cruz-Ramírez, M., Hervás-Martínez, C., Sánchez-Monedero, J., Gutiérrez, P.A.: Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. Neurocomputing **135**, 21–31 (2014). doi:10.1016/j.neucom.2013.05.058
19. Daniels, H., Velikova, M.: Derivation of monotone decision models from noisy data. IEEE Trans. Syst. Man Cybern. Part C **36**, 705–710 (2006)
20. Daniels, H., Velikova, M.: Monotone and partially monotone neural networks. IEEE Trans. Neural Netw. **21**(6), 906–917 (2010)
21. Dembczyński, K., Kotłowski, W., Słowiński, R.: Learning rule ensembles for ordinal classification with monotonicity constraints. Fundamenta Informaticae **94**(2), 163–178 (2009)
22. Duivesteijn, W., Feelders, A.: Nearest neighbour classification with monotonicity constraints. ECML/PKDD **1**, 301–316 (2008)
23. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. **27**, 861–874 (2006)
24. Feelders, A.: Monotone relabeling in ordinal classification. In: IEEE International Conference on Data Mining (ICDM), pp. 803–808 (2010)
25. Feelders, A.J., Pardoel, M.: Pruning for monotone classification trees. In: IDA, Lecture Notes in Computer Science, vol. 2810, pp. 1–12. Springer, New York (2003)
26. Fernández-Navarro, F., Riccardi, A., Carloni, S.: Ordinal neural networks without iterative tuning. IEEE Trans. Neural Netw. Learn. Syst. **25**(11), 2075–2085 (2014)
27. Fullerton, A.S., Xu, J.: The proportional odds with partial proportionality constraints model for ordinal response variables. Soc. Sci. Res. **41**(1), 182–198 (2012). doi:10.1016/j.ssresearch.2011.09.003
28. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **42**(4), 463–484 (2012)
29. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes. Pattern Recognit. **44**(8), 1761–1776 (2011). doi:10.1016/j.patcog.2011.01.017
30. García, J., AlBar, A.M., Aljohani, N.R., Cano, J.R., García, S.: Hyperrectangles selection for monotonic classification by using evolutionary algorithms. Int. J. Comput. Intell. Syst. **9**(1), 184–202 (2016)
31. García, J., Fardoun, H.M., Alghazzawi, D.M., Cano, J.R., García, S.: MoNGEL: monotonic nested generalized exemplar learning. Pattern Anal. Appl. (2016) (In press). doi:10.1007/s10044-015-0506-y
32. García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining. Springer, New York (2015)
33. González, S., Herrera, F., García, S.: Monotonic random forest with an ensemble pruning mechanism based on the degree of monotonicity. New Gener. Comput. **33**(4), 367–388 (2015)
34. Gutiérrez, P.A., Pérez-Ortiz, M., Sánchez-Monedero, J., Fernandez-Navarro, F., Hervás-Martínez, C.: Ordinal regression methods: survey and experimental study. IEEE Trans. Knowl. Data Eng. **28**(1), 127–146 (2016). doi:10.1109/TKDE.2015.2457911

35. Harrington, E.F.: Online ranking/collaborative filtering using the perceptron algorithm. In: Proceedings of the Twentieth International Conference on Machine Learning (ICML2003) (2003)
36. Hu, Q., Che, X.Z.L., Zhang, D., Guo, M., Yu, D.: Rank entropy-based decision trees for monotonic classification. IEEE Trans. Knowl. Data Eng. **24**(11), 2052–2064 (2012)
37. Hühn, J.C., Hüllermeier, E.: Is an ordinal class structure useful in classifier learning? Int. J. Data Mining Model. Manag. **1**(1), 45–67 (2008)
38. van de Kamp, R., Feelders, A., Barile, N.: Isotonic classification trees. In: Advances in Intelligent Data Analysis VIII, Proceedings of 8th International Symposium on Intelligent Data Analysis, IDA 2009, Lyon, France, August 31–September 2, 2009, pp. 405–416 (2009)
39. Kotlowski, W., Slowinski, R.: Rule learning with monotonicity constraints. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14–18, 2009, pp. 537–544 (2009)
40. Kotlowski, W., Słowiński, R.: On nonparametric ordinal classification with monotonicity constraints. IEEE Trans. Knowl. Eng. **25**(11), 2576–2589 (2013)
41. Lee, J.W.T., Yeung, D.S., Wang, X.: Monotonic decision tree for ordinal classification. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 2623–2628 (2003)
42. Lievens, S., De Baets, B., Cao-Van, K.: A probabilistic framework for the design of instance-based supervised ranking algorithms in an ordinal setting. Ann. Oper. Res. **163**(1), 115–142 (2008)
43. Lin, H.T., Li, L.: Reduction from cost-sensitive ordinal ranking to weighted binary classification. Neural Comput. **24**(5), 1329–1367 (2012)
44. Makino, K., Suda, T., Ono, H., Ibaraki, T.: Data analysis by positive decision trees. IEICE Trans. Inf. Syst. **E82**(D(1)), 76–88 (1999)
45. Marsala, C., Petturiti, D.: Rank discrimination measures for enforcing monotonicity in decision tree induction. Inf. Sci. **291**, 143–171 (2015)
46. McCullagh, P.: Regression models for ordinal data. J. R. Stat. Soc. Ser. B (Methodological) **42**(2), 109–142 (1980)
47. Milstein, I., Ben-David, A., Potharst, R.: Generating noisy monotone ordinal datasets. Artif. Intell. Res. **3**(1), 30–37 (2014)
48. Nathalie Japkowicz, M.S.: Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press, Cambridge (2011)
49. Novak, P.K., Lavrac, N., Webb, G.I.: Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. J. Mach. Learn. Res. **10**, 377–403 (2009)
50. PASCAL: Pascal (Pattern Analysis, Statistical Modelling and Computational Learning) Machine Learning Benchmarks Repository (2011). http://mldata.org/
51. Perez-Ortiz, M., Gutierrez, P.A., Hervas-Martinez, C., Yao, X.: Graph-based approaches for over-sampling in the context of ordinal regression. IEEE Trans. Knowl. Data Eng. **27**(5), 1233–1245 (2015)
52. Potharst, R., Ben-David, A., van Wezel, M.C.: Two algorithms for generating structured and unstructured monotone ordinal datasets. Eng. Appl. Artif. Intell. **22**(4–5), 491–496 (2009)
53. Potharst, R., Feelders, A.J.: Classification trees for problems with monotonicity constraints. SIGKDD Explor. **4**(1), 1–10 (2002)
54. Qian, Y., Xu, H., Liang, J., Liu, B., Wang, J.: Fusing monotonic decision trees. IEEE Trans. Knowl. Data Eng. **27**(10), 2717–2728 (2015)
55. Rademaker, M., De Baets, B., De Meyer, H.: Optimal monotone relabelling of partially non-monotone ordinal data. Optim. Methods Softw. **27**(1), 17–31 (2012)
56. Sousa, R.G., Cardoso, J.S.: Ensemble of decision trees with global constraints for ordinal classification. In: 11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22–24, 2011, pp. 1164–1169 (2011)
57. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel discriminant learning for ordinal regression. IEEE Trans. Knowl. Data Eng. **22**(6), 906–910 (2010)
58. Torra, V., Domingo-Ferrer, J., Mateo-Sanz, J.M., Ng, M.: Regression for ordinal variables without underlying continuous variables. Inf. Sci. **176**(4), 465–474 (2006)
59. Triguero, I., García, S., Herrera, F.: Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl. Inf. Syst. **42**(2), 245–284 (2015)
60. Van Gestel, T., Baesens, B., Van Dijcke, P., Garcia, J., Suykens, J., Vanthienen, J.: A process model to develop an internal rating system: Sovereign credit ratings. Decision Support Syst. **42**(2), 1131–1151 (2006)
61. Waegeman, W., De Baets, B., Boullart, L.: Roc analysis in ordinal regression learning. Pattern Recognit. Lett. **29**(1), 1–9 (2008)
62. Williams, R.: Generalized ordered logit/partial proportional odds models for ordinal dependent variables. Stata J. **6**(1), 58–82 (2006)
63. Zhu, H., Zhai, J., Wang, S., Wang, X.: Monotonic decision tree for interval valued data. In: 13th International Conference on Machine Learning and Cybernetics, pp. 231–240 (2014)