Neurocomputing 341 (2019) 168–182

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Monotonic classification: An overview on algorithms, performance measures and data sets

José-Ramón Cano[a], Pedro Antonio Gutiérrez[b], Bartosz Krawczyk[c], Michał Woźniak[d], Salvador García[e,*]

[a] Department of Computer Science, University of Jaén, EPS of Linares, Avenida de la Universidad S/N, Linares 23700, Jaén, Spain
[b] Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
[c] Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA
[d] Department of Computer Science, Wrocław University of Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
[e] Department of Computer Science and Artificial Intelligence, University of Granada, Granada 18071, Spain

ARTICLE INFO

Article history:
Received 17 November 2018
Revised 4 February 2019
Accepted 11 February 2019
Available online 14 March 2019

Communicated by Dr. Nianyin Zeng

Keywords:
Monotonic classification
Ordinal classification
Taxonomy
Software
Performance metrics
Monotonic data sets

ABSTRACT

Currently, knowledge discovery in databases is an essential first step when identifying valid, novel and useful patterns for decision making. There are many real-world scenarios, such as bankruptcy prediction, option pricing or medical diagnosis, where the classification models to be learned need to fulfill restrictions of monotonicity (i.e. the target class label should not decrease when input attributes values increase). For instance, it is rational to assume that a higher debt ratio of a company should never result in a lower level of bankruptcy risk. Consequently, there is a growing interest from the data mining research community concerning monotonic predictive models. This paper aims to present an overview of the literature in the field, analyzing existing techniques and proposing a taxonomy of the algorithms based on the type of model generated. For each method, we review the quality metrics considered in the evaluation and the different data sets and monotonic problems used in the analysis. In this way, this paper serves as an overview of monotonic classification research in specialized literature and can be used as a functional guide for the field.

1. Introduction

Data mining, as a key stage in the discovery of knowledge, is aimed at extracting models that represent data in ways we may not have previously taken into consideration [1]. Among all the data mining alternatives, we focus our attention on classification as a predictive task [2,3]. There is a particular case of predictive classification where the target class takes values in a set of ordered categories. In the case at hand, we are referring to ordinal classification or regression [4]. In addition, the classification task is defined as monotonic classification in those cases in which domains of attributes have been ordered and a monotonic relationship exists between an evaluation of an object in the attributes and its class assignment [5].

Monotonicity is a type of background knowledge of vital importance for many real problems, which is needed to obtain more accurate, robust and fairer models of the data considered. In this way, monotonicity can be found in different environments such as economics, natural language or game theory [5], as well as the evaluation of courses at teaching institutions [6].

Some important examples of real problems where this kind of background knowledge has to be considered are being analyzed today. For bankruptcy prediction in companies in time [7], appropriate action should be taken considering the information based on financial indicators taken from their annual reports. Monotonicity is present in the comparison of two companies where one dominates the other in all financial indicators. Because of this dominance, the overall evaluation of the second one should not be higher than that of the first. In this way, monotonic classification has been applied to predict the credit rating score used by banks [8]. Another example is the house pricing problem [9], in which we should assure that the price of a house increases with an increase of the number of rooms or with the availability of air conditioning, and that it decreases with, for example, the pollution concentration in the area.

Considering monotonicity constraints in a learning task is motivated by two main facts [10]: (1) the size of the hypothesis space

* Corresponding author.
E-mail addresses: jrcano@ujaen.es (J.-R. Cano), pagutierrez@uco.es (P.A. Gutiérrez), bkrawczyk@vcu.edu (B. Krawczyk), michal.wozniak@pwr.edu.pl (M. Woźniak), salvagl@decsai.ugr.es (S. García).

https://doi.org/10.1016/j.neucom.2019.02.024
0925-2312/

which facilitates the learning process, is reduced; (2) other metrics besides accuracy, such as the consistency with respect to these constraints, can be used by experts to accept or reject certain models.

In this way, the need of handling background knowledge about ordinal evaluations and monotonicity constraints in the learning process has led to the development of new algorithms. The interest in the field of monotonic classification has significantly increased [11,12], leading to a growing number of techniques and methods. Apart from these algorithmic developments, different quality measures have been presented to measure the consistency with respect monotonicity constraints.

Given that, to the knowledge of the authors, there are no functional guides for this domain of study, and it can be difficult to obtain a general overview of the state of the art. Because of this reason, this paper presents an overview on the monotonic classification field, including:

- A systematic review of the techniques proposed in the literature.
- A taxonomy to categorize all the existing algorithms, including whether or not there is publicly available software related to them.
- The quality measures applied to evaluate the performance of monotonic classifiers in the literature. These metrics analyze the performance both in terms of accuracy and degree of fulfillment of the monotonicity constraints.
- Finally, the data sets considered in every proposal and a summary of which are the most used and where they can be found.

The remainder of this paper is structured as follows. Section 2 presents a definition of the monotonic classification problem. Section 3 shows an overview of the monotonic methods and the taxonomy proposed to categorize them. Section 4 offers an analysis of the quality metrics considered in monotonic classification. Section 5 presents the data sets evaluated in the literature, highlighting the most popular ones and where they can be found. In Section 6 we offer some guidelines regarding existing methods to researchers interested in this topic and we enumerate some recommendations for future research. Finally, Section 7 is devoted to the conclusions reached.

2. Definition of monotonic classification

The process of data knowledge discovery in databases is a key objective for organizations to make accurate and timely decisions and recognize the value in data sources. One of the main stages within the process is data mining [1], where models are extracted from the input data collected. These models are used to support people in making decisions about problems that may be rapidly changing and not easily specified in advance (i.e. unstructured and semi-structured decision problems). Among all kinds of models, we focus our attention on classification algorithms, where the goal is to predict the value of a target variable. When the target variable exhibits a natural ordering, we are talking about ordinal classification (also known as ordinal regression) [4,11,13,14]. The order of the categories can be exploited to construct more accurate models in those application domains involving preferences, like social choice, multiple criteria decision making, or decision under risk and uncertainty. For example, in a factory a worker can be evaluated as “excellent”, “good” or “bad”, or a credit risk can be rated as “AAA”, “AA”, “A” or “A-”. A particular case of ordinal classification is monotonic classification [11]. The interest in monotonic classification of the scientific community has increased over the last few years. This fact can be corroborated in Fig. 1, where the number of proposals in the specialized literature is represented over time.

Classification problems where there is background knowledge in the form of ordinal evaluations and monotonicity constraints are very common. In this kind of problem, the order properties of the input space are exploited, by using the available knowledge in terms of dominance relation (one sample dominates another when each coordinate of the former is not smaller than the respective coordinate of the latter). Monotonicity constraints require that the class label assigned to a pattern should be greater or equal to the class labels assigned to the patterns it dominates. As an example, consider a monotonicity constraint relating one input attribute and the target class. In this case, a sample in the data set with a higher value of the input attribute should not be associated to a lower class value, as long as the other attributes of the sample are fixed. A monotonicity constraint always involves one input attribute and the class attribute, and there should be, at least, one monotonicity constraint (to distinguish monotonic classification from ordinal regression). Monotonicity constraints can be either direct (as the example presented before) or inverse (if the value of the attribute decreases, the class value should not increase). Usually, in real monotonic classification problems, the monotonicity constraints are assumed only for a subset of the input features.

As a descriptive example, we can consider student evaluation in a college, the students being evaluated with a rating between 0 and 10. We consider three students (Student A, B and C) with 22 evaluations each one and a final mark. We consider that all the input attributes (22 evaluations) have a direct monotonic assumption with respect to the output value (final qualification, represented in bold face):

- Student A: 5,5,5,5,7,6,5,5,5,5,5,5,6,5,5,6,6,6,5,5,5,5,**4**.
- Student B: 3,5,3,4,7,3,3,5,3,3,3,3,6,3,3,4,3,6,4,3,5,3,**5**.
- Student C: 2,2,1,2,1,2,2,3,2,2,1,2,3,2,2,3,3,2,2,1,2,3,**2**.

As can be observed, there is a monotonic violation involving two samples (Students A and B), where Student B, who has worse or equal evaluation marks than Student A, who presents a higher final qualification. On the other hand, there are no monotonic violations when considering Student C with respect to both Students A and B.

Now, we formally define a classification data set with ordinal labels and monotonicity constraints. Let us assume that patterns are described using a total of f input variables with ordered domains, $\mathbf{x}_i \subseteq \mathbb{R}^f$, and a class label, y_i, from a finite set of C ordered labels, $y_i \in \mathcal{Y} = \{1, \ldots, C\}$. In this way, the data set D consists of n samples or instances $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. As previously discussed, a *dominance relation*, $\succeq$, is defined as follows:

$$\mathbf{x} \succeq \mathbf{x}' \Leftrightarrow x^s \geq x^{s'} \forall s \text{ with a monotonicity constraint}, \tag{1}$$

where x^s and $x^{s'}$ are the sth coordinates of patterns $\mathbf{x}$ and $\mathbf{x}'$, respectively. In other words, $\mathbf{x}$ dominates $\mathbf{x}'$ if each coordinate of $\mathbf{x}$ is not smaller than the respective coordinate of $\mathbf{x}'$.

Samples $\mathbf{x}$ and $\mathbf{x}'$ in space D are *comparable* if either $\mathbf{x}' \succeq \mathbf{x}'$ or $\mathbf{x}' \succeq \mathbf{x}$. Both $\mathbf{x}$ and $\mathbf{x}'$ are *incomparable* otherwise. Two examples $\mathbf{x}$ and $\mathbf{x}'$ are *identical* if $x^j = x^{j'}, \forall j \in \{1, \ldots, f\}$, and they are *non-identical* if $\exists j$ for which $x^j \neq x^{j'}$.

A pair of comparable examples ($\mathbf{x}$, y) and ($\mathbf{x}'$, y') is said to be monotone if:[1]

$$\mathbf{x} \succeq \mathbf{x}' \wedge \mathbf{x} \neq \mathbf{x}' \wedge y \geq y', \tag{2}$$

or

$$\mathbf{x} = \mathbf{x}' \wedge y = y'. \tag{3}$$

[1] Recall that $y, y' \in \mathcal{Y} = \{1, \ldots, C\}$, so that every two labels can be compared using the ordinal scale.

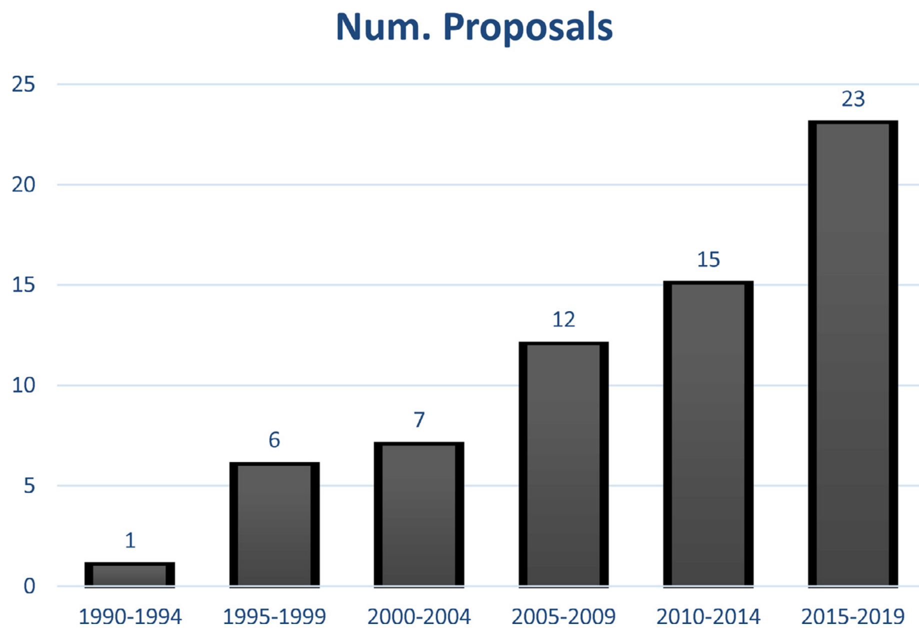


Fig. 1. Number of monotonic classification proposals over time.

A data set D with n examples is monotone if all possible pairs of examples are either monotone or incomparable. It is worth mentioning that the previous notation was expressed for direct monotonicity constraints, but it could be changed to consider inverse ones. This definition considers that all f characteristics to be monotonous, forming a *fully monotone data set*.

However, in real life there may be data sets with monotonic (m) and non-monotonic (p) characteristics, forming a *partially monotone data set* whose definition would be as follows:

For $D = \{(\mathbf{x}_1^m, \mathbf{x}_1^p, y_1), \ldots, (\mathbf{x}_n^m, \mathbf{x}_n^p, y_n)\}$, where the patterns are described using f input variables ($f = m + p$), $\mathbf{x}_i^m \subseteq \mathbb{R}^{f_m}$ with ordered domains, $\mathbf{x}_i^p \subseteq \mathbb{R}^{f_p}$ with unordered domains and a class label, y_i, from a finite set of C ordered labels, $y_i \in \mathcal{Y} = \{1, \ldots, C\}$. The monotone partial order $\succeq_\eta$ is defined in expression (4) and partial monotonic data set in expression (5), for (($\mathbf{x}^m$, $\mathbf{x}^p$), y) and (($\mathbf{x}^{m'}$, $\mathbf{x}^{p'}$), y'):

$$(\mathbf{x}^m, \mathbf{x}^p) \succeq_\eta (\mathbf{x}^{m'}, \mathbf{x}^{p'}) \Leftrightarrow x^m \geq x^{m'} \forall m,\ x^p = x^{p'} \forall p \tag{4}$$

$$(\mathbf{x}^m, \mathbf{x}^p) \succeq_\eta (\mathbf{x}^{m'}, \mathbf{x}^{p'}) \wedge y \geq y',\ \forall (\mathbf{x}^m, \mathbf{x}^p), (\mathbf{x}^{m'}, \mathbf{x}^{p'}) \in D \tag{5}$$

3. A taxonomy for monotonic classification algorithms

This section presents and describes the proposals in the specialized literature for monotonic classification, deriving a taxonomy from them.

The Knowledge Data Discovery process is composed of several stages. Two of them are usually known as data preprocessing and data mining [15]. In monotonic classification, the algorithms present in the literature belong to one of these two stages: data preprocessing [16] for monotonic classification problems (here, we denote it as Monotonic Data Preprocessing) or knowledge extraction through monotonic classification [11], respectively. The remaining categorizations are based on the goal of the different methods, the heuristics followed and the models generated by each algorithm. In this sense, the algorithms proposed can be divided into:

1. Monotonic Classifiers, aiming at the generation of predictive models satisfying the monotonicity constraints either partially or totally. There are several families of classifiers depending on the type of model they build:
   - Instance based classifiers. These algorithms do not build a model but they directly use the instances of the data set of to make classification decisions.
   - Decision trees or classification rules. In this case, the models built involve readable production rules in forms of decision trees or a set of rules.
   - Ensembles [17] or multiclassifiers. This group is composed by methods which use several classifiers to obtain different responses, which are aggregated into a global classification decision. Two classical approaches are considered:
     - Boosting: a number of weak learners are combined to create a strong classifier able to achieve accurate predictions. These algorithms use all data to train each learner, but the instances are associated with different weights representing their relevance in the learning process. If an instance is misclassified by a weak learner, its weight is increased so that subsequent learners focus on them. This process is applied iteratively.
     - Bagging: it chooses random subsets of samples with replacement of the data set, and a (potentially) weak learner is trained from each subset.
   - Neural Networks. These are biologically inspired models, where the function relating inputs and target attribute consists of a set of building blocks (neurons), which are organized in layers and interconnected. An iterative training process is performed to obtain the values of connection weights. They are the precursors of Deep Learning, which is currently the most promising area in Machine Learning [18].
   - Support Vector Machines. This family considers support vector machines based learning and derivatives.
   - Hybrid. This last set of algorithms considers the combination of different classification algorithms into a hybrid one (for example, rule and instance-based learning).
   - Fuzzy Integral. These algorithms are based on the use of the Choquet integral which can be seen as a

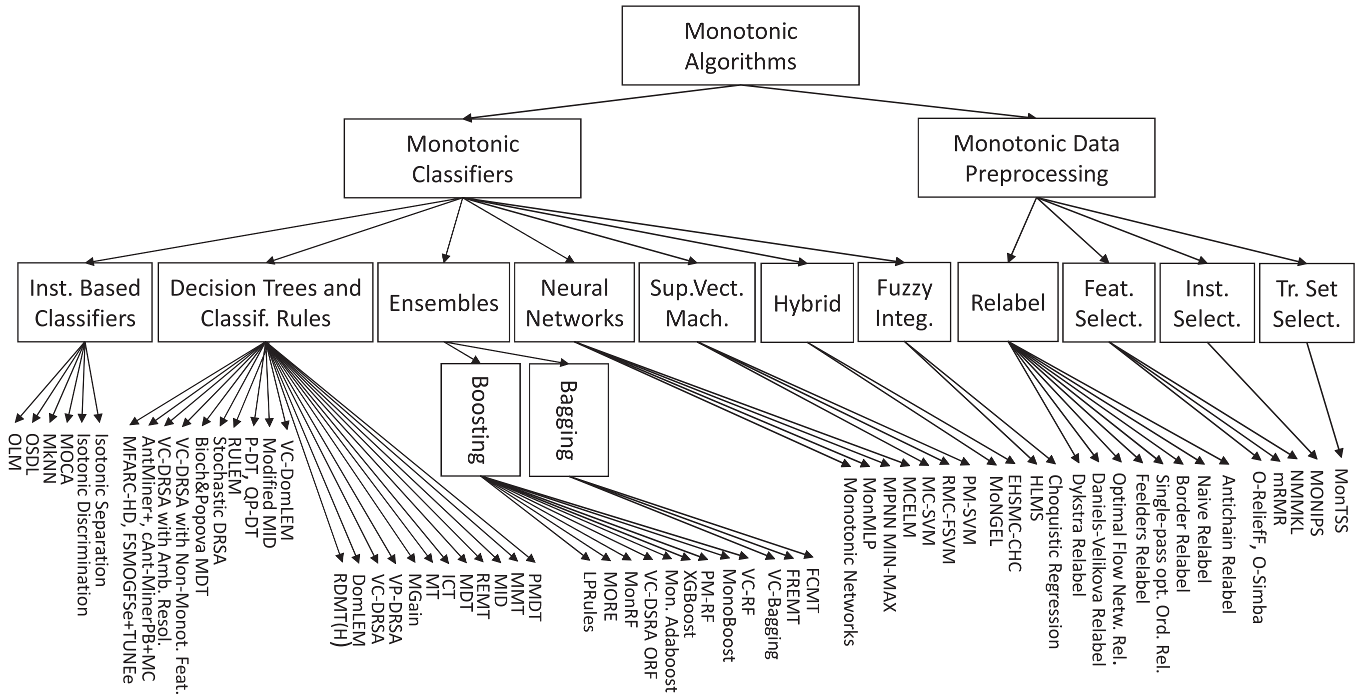


Fig. 2. Monotonic algorithms taxonomy.

generalization of the standard (Lebesque) integral to the case of non-additive measures [19].

2. Monotonic Data Preprocessing refines the data sets in order to improve the performance of monotonic classification algorithms:
   - Relabeling. These methods change the label of the instances to minimize the number of monotonicity violations present in the data set.
   - Feature selection. Their objective is to obtain the most relevant features to improve monotonic classification performance.
   - Instance selection. In this case, a subset of samples is selected from the data set with the objective of deriving better monotonic classifiers.
   - Training set selection. The heuristic followed by this set of algorithms must be generic in such a way that the selected set is the one that reports the highest performance regardless of the classifier subsequently used.

Fig. 2 shows the proposed taxonomy and Tables 1 and 2 the summary of all the monotonic classifiers found in the specialized literature. The first column of the table contains the year of the proposal, the second is the reference and the third is the proposal name. We also show in the fourth and fifth columns, whether or not the algorithm requires a total monotonic input data set and whether or not it produces complete monotonic output models, respectively. The sixth column indicates whether the algorithm accepts partially monotonic data sets [20]. Seventh and eighth columns present the non monotonic classification algorithms used as a baseline to compare the method and the monotonic classifiers used for comparison in the experimental analysis conducted in each paper. The last column shows whether or not the algorithm's source code is publicly available and, if it is, the name of framework in which we can find it. All algorithms are capable of dealing with multiclass problems, except for one of them which will be indicated in its description.

Next, we provide a description of the methods in each family.

3.1. Monotonic classifiers

3.1.1. Instance based classifiers

- Ordered Learning Model (*OLM* [10,69]). New objects are classified by the following function:

$$f_{\text{OLM}}(\mathbf{x}) = \max\{\mathbf{y}_i : (\mathbf{x}_i, \mathbf{y}_i) \in D, \mathbf{x}_i \preceq \mathbf{x}\}. \tag{6}$$

  If there is no object from D which is dominated by $\mathbf{x}$, then a class label is assigned by a nearest neighbor rule. D is chosen to be consistent and not to contain redundant examples. An object $(\mathbf{x}_i, \mathbf{y}_i)$ is redundant in D if there is another object $(\mathbf{x}_j, \mathbf{y}_j)$ such that $\mathbf{x}_i \succeq \mathbf{x}_j$ and $\mathbf{y}_i = \mathbf{y}_j$.
- *Isotonic discrimination* [26]. This method applies isotonic regression based relabeling. After that, the limiting cumulative probability distribution for a prediction is evaluated considering the changes produced in the previous stage.
- *Isotonic separation* [32]. As a continuation of a relabeling process based on linear programming, instances that do not belong to boundaries are eliminated. The resulting boundaries are used to make the predictions.
- Ordered Stochastic Dominance Learner (*OSDL* [35,70]). For each sample $\mathbf{x}_i$, OSDL computes two mapping functions: one that is based on the examples that are stochastically dominated by $\mathbf{x}_i$ with the maximum label (of that subset), and the second is based on the examples that cover (i.e., dominate) $\mathbf{x}_i$, with the smallest label. Later, an interpolation between the two class values (based on their position) is returned as a class.
- Monotonic k-Nearest Neighbor (*MkNN* [36]). This classifier is an adaptation of the well-known nearest neighbor classifier, considering a full monotone data set. Starting from the original nearest neighbor rule, the class label assigned to a new data point $\mathbf{x}_0$ must lie in the interval $[y_{\min}, y_{\max}]$, where:

$$y_{\min} = \max\{\mathbf{y} | (\mathbf{x}, \mathbf{y}) \in D \wedge \mathbf{x} \preceq \mathbf{x}_0\}, \tag{7}$$

  and:

$$y_{\max} = \min\{\mathbf{y} | (\mathbf{x}, \mathbf{y}) \in D \wedge \mathbf{x}_0 \preceq \mathbf{x}\}. \tag{8}$$

Table 1
Monotonic classification methods reviewed. Part I.

Year	Reference	Abbr. name	Require Input monot.	Completely Monot. output	Partial Monot.	Comparison versus Classical methods	Comparison versus Monotonic methods	Code available in
1992	[10]	OLM	Yes	No	No	C4, ID3	None	[21] in WEKA
1995	[22]	MID	No	No	Yes	ID3	OLM	Not available
1995	[23]	HLMS	No	Yes	No	None	None	Not available
1997	[24]	Monotonic networks	Yes	Yes	No	None	None	Not available
1999	[25]	P-DT, QP-DT	Yes, No	Yes, No	No, No	ID3	MID	Not available
1999	[26]	Isotonic discrimination	No	Yes	No	None	None	Not available
2000	[27]	MT	Yes	Yes	No	C4.5	OLM	Not available
2000	[28]	VC-DRSA	No	No	No	None	None	Not available
2000	[29]	DomLEM	No	No	No	None	None	Not available
2002	[30]	Bioch&Popova MDT	No	Yes	No	None	None	Not available
2002	[9]	Modified MID	No	No	Yes	None	None	Not available
2003	[31]	MDT	Yes	Yes	No	CART	None	Not available
2005	[32]	Isotonic Separation	No	No	No	None	None	Not available
2005	[33]	MonMLP	Yes	Yes	No	None	None	In CRAN
2007	[34]	VC-DRSA with Ambig. Resol.	No	No	No	None	None	Not available
2008	[35]	OSDL	No	Yes	No	None	None	[21] in WEKA
2008	[36]	MkNN	No	Yes	No	kNN	None	Not available
2008	[37]	MOCA	No	Yes	No	OSDL	None	Not available
2008	[38]	Stochastic DRSA	No	No	No	None	None	Not available
2009	[39]	ICT	No	Yes	Yes	None	None	Not available
2009	[40]	LPRules	No	Yes	No	J48, SVM	OLM, ICT	Not available
2009	[41]	VP-DRSA	No	No	No	None	None	Not available
2009	[42]	MORE	No	Yes	No	SVM, J48, kNN	None	Not available
2010	[20]	MPNN MIN–MAX	No	No	Yes	None	None	Not available
2010	[43]	VC-bagging	No	No	No	None	OLM, OSDL	Not available
2011	[44]	VC-DomLEM	No	No	No	Naive Bayes, SVM, Ripper, C4.5	OLM, OSDL	Not available

Table 2
Monotonic classification methods reviewed. Part II.

Year	Reference	Abbr. name	Require Input monot.	Completely Monot. output	Partial Monot.	Comparison versus Classical methods	Comparison versus Monotonic methods	Code available in
2012	[45]	REMT	No	No	No	CART, Rank Tree	OLM, OSDL	Not Available
2012	[19]	Choquistic Regression	Yes	Yes	No	MORE	LMT, Logistic Regression	Not Available
2012	[46]	VC-DRSA with Non-Monot. Features	No	No	Yes	Naive Bayes, SVM, Ripper, C4.5, MODLEM	None	Not Available
2014	[8]	MC-SVM	Yes	Yes	No	SVM	None	Not Available
2015	[47]	MGain	No	No	No	C4.5	None	Not Available
2015	[48]	FREMT	No	No	No	None	REMT	Not Available
2015	[49]	MonRF	No	No	No	None	OLM, OSDL, MID	Not Available
2015	[50]	VC-DRSA ORF	No	No	No	None	None	[51] in jMAF
2015	[52]	RDMT(H)	No	No	No	None	MID, ICT	Not Available
2015	[53]	RMC-FSVM	No	No	No	FSVM, SVM	None	Not Available
2015	[54]	VC-RF	No	No	No	None	VC-DRSA with Non-Monot. Feat., VC-DomLEM	Not Available
2016	[55]	MoNGEL	No	No	Yes	None	MkNN, OLM, OSDL	[56] in Java
2016	[57]	Monot. AdaBoost	No	No	Yes	None	MID	Not Available
2016	[58]	AntMiner+, cAnt-Miner$_{PB}$+MC	No, No	Yes, Yes	Yes, Yes	ZeroR	OLM	Not Available
2016	[59]	EHSMC-CHC	No	No	No	None	MkNN, OLM, OSDL, MID	Not Available
2016	[60]	XGBoost	No	Yes	Yes	pGBRT, Spark MLLib, H2O	None	[60] in GitHub
2016	[61]	PM-SVM	No	No	Yes	SVM	MC-SVM	[61] in GitHub
2016	[62]	PM-RF	No	No	Yes	Random Forest	MC-SVM	[62] in GitHub
2016	[63]	MMT	No	No	Yes	ID3, J48, CART, RandomTree	REMT, OLM, OSDL, RDMT(H)	Not Available
2017	[64]	FCMT	No	No	No	REMT, FREMT	None	Not Available
2017	[12]	MCELM	No	Yes	No	CART, Rank Tree, ELM	OLM, OSDL, REMT	Not Available
2017	[65]	RULEM	No	Yes	Yes	Ripper, C4.5	AntMiner+	Not Available
2017	[66]	MFARC-HD, FS$_{MOGFS^e}$+T$_{UN^e}$	No, No	No, No	No, No	WM	OSDL, MkNN, C4.5-MID, OLM, EHSMC-CHC, RF-MID	Not Available
2018	[67]	MonoBoost	No	No	Yes	kNN	None	[67] in GitHub
2018	[68]	PMDT	No	No	Yes	None	REMT, OLM, OSDL, RDMT(H)	Not Available

- *MOCA* [37]. MOCA is a nonparametric monotone classification algorithm that attempts to minimize the mean absolute prediction error for classification problems with ordered class labels. Firstly, the algorithm obtains a monotone classifier considering only training data. In the test phase, a simple interpolation scheme is applied.

3.1.2. Decision trees and classification rules

- Monotonic Induction of Decision trees (*MID* [22]). Ben-David introduces a measure of non-monotonicity in the classical classification decision tree ID3 algorithm [71]. This measure was denoted as total-ambiguity-score. To calculate it, a non-monotonicity $b \times b$ matrix M must be constructed, related to a tree containing b branches. Each value m_{ij} is 1 if the branches i and j are non-monotone, and 0 if they are.

- Positive Decision Tree, Quasi-Positive Decision Tree (*P-DT, QP-DT* [25]). In these algorithms the splitting rule separates the points that have the right child-node larger than the left child-node (in the sense of the target variable). The algorithm adds samples to the nodes in such way that the resulting tree is monotone. This algorithm requires as a precondition to be applied on strictly monotone binary data sets, containing only two classes. This is the only method which is not able to deal with multiclass data sets.
- Variable Consistency model of Dominance-based Rough Sets Approach (*VC-DRSA* [28]). The method introduces a relaxation to the DRSA model, which admits some inconsistent objects to the lower approximations; the relaxation is controlled by an index called consistency level. VC-DRSA is insensitive to marginal inconsistencies which appear in data sets.
- Monotonic Tree (*MT* [27]). Potharst and Bioch present a tree generation algorithm for monotonic classification problems with discrete domains for multiclass data sets. In addition, the proposal can be used to repair non-monotonic decision trees that have been generated by other methods.
- *DomLEM* [29]. This algorithm generates a complete and non-redundant set of decision rules, heuristically tending to minimize the number of rules generated. It is able to produce decision rules accepting a limited number of negative examples within the variable consistency model of the dominance rough sets approach.
- *Modified MID* [9]. In this case, an improvement of the order ambiguity in MID algorithm is proposed by the authors. The new order ambiguity weighs nonmonotone leaf pairs by the probability of leaf appearance.
- Bioch&Popova Monotone Decision Tree (*Bioch&Popova MDT* [30]). This algorithm generates monotonic decision trees from noisy data modifying the update rule. It controls the size of the trees by means of pre- and post-pruning while the tree is guaranteed to remain monotone.
- Monotonic Decision Tree (*MDT* [31]). The authors proposed an induction approach to generate monotonic decision trees from sets of examples which may not be monotonic or consistent. The algorithm constructs the tree using a set of ordinal labels which are not the same as the original ones. A mapping process can be used to relabel them into the originals.
- *VC-DRSA with Ambiguity Resolution* [34]. This method induces the rules from rough approximations of preference-ordered decision classes, according to Variable Consistency Dominance-based Rough Set Approach. When ambiguity appears in the prediction of the class of a new instance to evaluate, the method assigns a given instance to a class characterized by a maximum positive difference between strength of rule premises suggesting assignment to this class and those discouraging such an assignment.
- *Stochastic DRSA* [38]. The proposal presents a new stochastic approach to dominance-based rough sets, whose application results in estimating the class interval for each instance. The class interval generated has the form of a confidence interval and follows from the empirical risk minimization of the specific loss function.
- Variable Precision Dominance-based Rough Set Approach (*VP-DRSA* [41]). The authors offers a proposal to treat errors in the framework of DRSA. They introduce the concept of variable precision rough set approach.
- Isotonic Classification Tree (*ICT* [39]). This approach adjusts the probability estimated in the leaf nodes in case of a monotonicity violation. The idea is that, considering the monotonicity constraint, the sum of the absolute prediction errors on the training sample should be minimized. In addition, this algorithm can also handle problems where some, but not all, attributes have a monotonic relation with respect to the response.
- Variable Consistency DomLEM (*VC-DomLEM* [44]). Improvement of the DomLEM method to transfer the probabilistic characteristic of variable consistency approaches through to rule induction.
- *VC-DRSA with Non-Monotone Features* [46]. The relationships in the data are represented by monotonic decision rules. To discover the monotonic rules, the authors propose a non-invasive transformation of the input data, and a way of structuring them into consistent and inconsistent parts using VC-DRSA.
- Rank Entropy based Monotonic decision Trees (*REMT* [45]). This algorithm introduces a metric called rank entropy as a robust measure of feature quality. It is used to compute the uncertainty, reflecting the ordinal structures in monotonic classification. The construction of the decision tree is based on this measure.
- *RDMT(H)* [52]. Marsala and Petturiti presented a tree classifier parameterized by a discrimination measure H, which is considered for splitting, together with other three pre-pruning parameters. RDMT(H) guarantees a weak form of monotonicity for the resulting tree when the data set is monotone consistent and H refers to any rank discrimination measure. The authors adapted different measures to monotonic classification.
- *MGain* [47]. MGain introduces the index of the monotonic consistency of a cut point with respect to a data set. When non-monotonic data appear in the training set, the index of monotonic consistency selects the best cut point. If the initial data set is totally monotonic, the results obtained are similar to those using C4.5 [72].
- *AntMiner+, cAnt-Miner*$_{PB}$*+MC* [58]. These algorithms are an extension of an existing ant colony optimization based classification rule learner, able to create lists of monotonic classification rules. They consider an improved sequential covering strategy to search for the best list of classification rules.
- Monotonic Multivariate Trees (*MMT* [63]). The proposed method discovers partitions via oblique hyperplane in the input space. MMT generates the projections of the objects which are used to split the data by improved splitting criteria with rank mutual information or rank Gini impurity.
- Rule Learning of ordinal classification with Monotonicity constraints (*RULEM* [65]). The authors present a technique to induce monotonic ordinal rule based classification models, which can be applied in combination with any rule or tree induction technique in a post processing step. They also introduce two metrics to evaluate the plausibility of the ordinal classification models obtained.
- *MFARC-HD* [66]. In this case, different mechanisms based on monotonicity indexes are coupled with a popular and competitive classification evolutionary fuzzy system: FARC-HD. In addition, the proposal is able to handle any kind of classification data set without a preprocessing step.
- $FS_{MOGFS^e}+T_{UN^e}$ [66]. The proposed method consists of two separated stages for learning and subsequent tuning. The first stage is based on an improved multi-objective evolutionary algorithm designed to select the relevant features while learning the appropriate granularities of the membership functions. In the second stage, an evolutionary post-process is applied to the knowledge base obtained.

- Partially Monotonic Decision Trees (*PMDT* [68]). The authors propose a rank-inconsistent rate that distinguishes attributes from criteria. That rate represents the directions of the monotonic relationships between criteria and decisions. Finally, a partially monotonic decision tree algorithm is designed to extract decision rules for partially monotonic classification tasks.

3.1.3. Ensembles

1. Boosting
   - *LPRules* [40]. This algorithm is based on a statistical analysis of the problem, trying to relate monotonicity constraints to the constraints imposed on the probability distribution. First, LPRules decomposes the problem into a sequence of binary subproblems. Then, the data for each subproblem is monotonized using a non-parametric approach by means of the class of all monotone functions. In the last step, a rule ensemble is generated using the LPBoost method to avoid errors in the monotonized data.
   - MOnotone Rule Ensembles (*MORE* [42]). MORE uses forward a stage-wise additive modeling scheme for generating an ensemble of decision rules for binary problems. An advantage of this method, as the authors indicate, is its comprehensibility and consistence.
   - Monotonic Random Forest (*MonRF* [49]). The method is an adaptation of Random Forest [73] for classification with monotonicity constraints, including the rate of monotonicity as a parameter to be randomized during the growth of the trees. An ensemble pruning mechanism based on the monotonicity index of each tree is used to select the subset of the most monotonic decision trees which constitute the forest.
   - Variable Consistency Dominance-based Rough Set Approach Ordinal Random Forest (*VC-DRSA ORF* [50]). The authors propose an Ordinal Random Forest based on the variable consistence dominance rough set approach. The ordinal random forest algorithm is implemented using Hadoop [74].
   - Variable Consistency Random Forest (*VC-RF* [54]). Wang et al. propose the dominance and fuzzy preference inconsistency rates, which have the capacity of discovering global monotonicity relationships directly from data rather than induced rules. The method includes a refined transformation, in which an additional step is introduced to determine whether an ordinal condition attribute should be cloned or not according to its inconsistency rates.
   - *Monotonic Adaboost* [57]. In this case, decision trees are combined in an Adaboost scheme [75], considering a simple ensemble pruning method based on the degree of monotonicity. The objective in this algorithm is to offer a good trade-off between accurate predictive performance and the construction of monotonic models.
   - *XGBoost* [60]. An open source library that provides the gradient boosting framework, which supports monotonic constraints as of version 0.71.
   - Partially Monotone Random Forest (*PM-RF* [62]). By creating a novel re-weighting scheme, PM-RF is an effective partially monotone approach that was particularly good at retaining accuracy while correcting highly non monotone data sets with many classes, albeit only achieving monotonicity locally.
   - *MonoBoost* [67]. Inspired by instance based classifiers, MonoBoost is a framework for monotone additive rule ensembles where partial monotonicity appears. The algorithm ensures perfect partial monotonicity with reasonable performance.

   There exist two publicly available and open source libraries that are absent from the literature: Arborist [76] and GBM [77]. Both are R packages that allow for monotone features by naïvely constraining each branch split (in each tree) to prohibit non monotone splits.
2. Bagging
   - Variable Consistency Bagging (*VC-bagging* [43]). For this proposal, the data set is structured using the Variable Consistency Dominance-based Rough Set Approach (VC-DRSA). A variable consistency bagging scheme is used to produce bootstrap samples that promote classification examples with relatively high consistency measure values.
   - Fusing Rank Entropy based Monotonic decision Trees (*FREMT* [48]). This method fuses decision trees taking into account attribute reduction and a fusing principle. The authors propose an attribute reduction approach with rank-preservation for learning base classifiers, which can effectively avoid overfitting and improve classification performance. In a second step, the authors establish a fusing principle considering the maximal probability by combining the base classifiers.
   - Fusing Complete Monotonic decision Trees (*FCMT* [64]). Xu et al. propose an improvement of FREMT algorithm using a discriminativeness matrix approach that guaranteed finding all satisfactory subsets.

3.1.4. Neural networks

- *Monotonic networks* [24]. Monotonic networks implements a piecewise-linear surface by taking maximum and minimum operations on groups of hyperplanes. Monotonicity constraints are enforced by constraining the sign of the hyperplane weight.
- Monotonic Multi-Layer Perceptron (*MonMLP* [33]). This algorithm satisfies the requirements of monotonicity for one or more inputs by constraining the sign of the weights of the multi-layer perceptron network. The performance of MonMLP does not depend on the quality of the training data because it is imposed in its structure.
- Monotonic Partial Neural Network MIN–MAX (*MPNN MIN–MAX* [20]). In this paper, the authors clarify some of the theoretical results on monotone neural networks with positive weights, which sometimes cause misunderstanding in the neural network literature. In addition, in the case of partially monotone problems they generalize the so-called MIN–MAX networks.
- Monotonic Classification Extreme Learning Machine (*MCELM* [12]). MCELM is a generalization of extreme learning machine for monotonic classification data sets. The proposal involves a quadratic programing problem in which the monotonicity relationships are considered to be constraints and the training errors as the objective to be minimized.

3.1.5. Support Vector Machines

- Monotonicity Constrained Support Vector Machine (*MC-SVM* [8,78]). MC-SVM is a rating model based on a support vector machine including monotonicity constraints in the optimization problem. The model is applied to credit rating, and the constraints are derived from the prior knowledge of financial experts.
- Regularized Monotonic Fuzzy Support Vector Machine (*RMC-FSVM* [53]). This method applies the Tikhonov regularization [79] to SVMs with monotonicity constraints in order to

ensure that the solution is unique and bounded. In this way, the prior domain knowledge of monotonicity can be represented in the form of inequalities based on the partial order of the training data.
- Partially Monotone Support Vector Machine (*PM-SVM* [61]). PM-SVM differs from the MC-SVM by proposing a new constraint generation technique designed to more efficiently achieve monotonicity.

3.1.6. Hybrid

- Monotonic Nested Generalized Exemplar Learning (*MoNGEL* [55]). MoNGEL combines instance-based and rule learning. The instances are converted to zero-dimensional rules, formed by a single point, obtaining an initial set of rules. As a second step, the method searches for that comparable rule of the same class with the minimum distance with respect to each rule, in order to iteratively generalize it. In the last step, the minimum number of non monotonic rules existing between them will be removed.
- Evolutionary Hyperrectangle Selection for Monotonic Classification (*EHSMC-CHC* [59]). After building a set of hyperrectangles from the training data set, a selection chosen by evolutionary algorithms is applied. In a preliminary stage, an initial set of hyperrectangles are generated by using a heuristic based on the training data, and then a selection process is carried out, focused on maximizing the performance considering several objectives, such as accuracy, coverage of examples and reduction of the monotonicity violations of the model with the lowest possible number of hyperrectangles.

3.1.7. Fuzzy Integrals

- Heuristic Least Mean Square (*HLMS* [23,80]). HLMS aims to identifying the fuzzy measure taking advantage of the lattice structure of the coefficients. Thanks to this identification, knowledge concerning the criteria can be obtained.
- *Choquistic Regression* [19,81,82]. The basic idea of choquistic regression is to replace the linear function of predictor variables, which is commonly used in logistic regression to model the log odds of the positive class, by the choquet integral [83].

3.2. Monotonic Data Preprocessing

Other group of methods in monotonic classification area are focused on applying data preprocessing techniques to improve the performance of monotonic classification algorithms [16]. So far the literature proposals follow four paths:

1. Relabeling. These methods aim at changing the class label of the instances which produce monotonicity violations to generate fully monotone data sets, which are required for many monotonic classifiers.
    - *Dykstra Relabel* [26]. These authors propose a monotone relabeling based on isotonic regression, able to minimize absolute error or squared error. The algorithm is optimal, optimizing those loss functions (absolute or squared error) but it does not guarantee the minimum number of label changes as it is not the key objective.
    - *Daniels–Velikova Greedy Relabel* [84,85]. This is a greedy algorithm used to relabel the non-monotone examples one at a time. At each step, it searches for the instance and the new label to maximize the increase in monotonicity of the data set. Although, at each step, it is able to maximize the jump towards complete monotonicity, the algorithm relabels more examples than is needed. This relabel method does not guarantee an optimal solution.
    - *Optimal Flow Network Relabel* [36,85,86]. This method is based on finding a maximum weight independent set in the monotonicity violation graph. Relabeling the complement of the maximum weight independent set results in a monotone data set with as few label changes as possible. This method is optimal, producing the minimal number of label changes.
    - *Feelders Relabel* [87–89]. This algorithm faces the problem of relabeling with minimal empirical loss as a convex cost closure problem. Feelders Relabel results in an optimal solution.
    - *Single-pass Optimal Ordinal Relabel* [90]. In this case, the idea is to exploit the properties of a minimum flow network and identify pleasing properties of some maximum cuts. As the name suggests, this is an optimal relabeling algorithm.
    - *Naive Relabel* [91]. This algorithm is a building block of the two algorithms detailed next, and uses a greedy scheme. The method does not guarantee an optimal solution.
    - *Border Relabel* [91]. This is a fast alternative to the greedy algorithm mentioned above, an it is more specific as it minimizes the deviations between the new and the original labels. This case is similar to the previous one, and thus is not optimal.
    - *Antichain Relabel* [91]. Based on the previous algorithm, this algorithm minimizes the total number of relabelings and leads to optimal solutions.
2. Feature Selection [92]. The objective of these methods is to improve the predictive capacity of the monotonic classifiers by selecting the most relevant characteristics.
    - *O-ReliefF, O-Simba* [93]
      The authors introduce margin-based feature selection algorithms for monotonic classification by incorporating the monotonicity constraints into the ordinal task. Relief and Simba methods are extended to the context of ordinal classification.
    - min-Redundancy Max-Relevance (*mRMR* [94–96])
      The algorithm mRMR integrates the rank mutual information metric with the search strategy of min-redundancy and max-relevance, creating an effective algorithm for monotonic feature selection.
    - Non-Monotonic feature selection via Multiple Kernel Learning (*NMMKL* [97]). Yang et al. propose a non-monotonic feature selection method that alleviates monotonic violations by computing the scores for individual features that depend on the number of selected features.
3. Instance Selection [98,99]. The idea behind these algorithms is to improve the performance of monotonic classifiers by selecting the most useful instances to be used as training set, using instance-based heuristics.
    - Monotonic Iterative Prototype Selection (*MONIPS* [6])
      MONIPS follows an iterative scheme in which it determines the most representative instances which maintain or improve the prediction capabilities of the MkNN algorithm. It follows an instance removal process based on the improvement of the MkNN performance.
4. Training Set Selection [100]. This set of algorithms has the same objective as those mentioned previously, except that the heuristic followed must be generic in such a way that the selected set is the one that reports the highest performance regardless of the classifier that is used on it later.

Table 3
Metrics considered in the reviewed monotonic classification methods.

Abbr. name	Predictive assessment metrics	Monotonicity fulfillment metrics
OLM	MSE	None
MID	MSE, MAE	NMI
HLMS	Accuracy	None
Monotonic networks	Error rate	None
P-DT, QP-DT	Error rate	None
Isotonic discrimination	None	None
MT	Accuracy	None
VC-DRSA	None	None
DomLEM	None	None
Bioch&Popova MDT	None	None
Modified MID	Error rate	NMI
MDT	Accuracy	γ_1, γ_2
Isotonic separation	None	None
MonMLP	None	None
VC-DRSA with amb. resol.	None	None
OSDL	None	None
MkNN	Error rate	None
MOCA	MAE	None
Stochastic DRSA	None	None
ICT	MAE	None
LPRules	MAE	None
VP-DRSA	None	None
MORE	MAE	None
MPNN MIN–MAX	MSE, error rate	None
VC-bagging	MAE	None
VC-DomLEM	MAE, accuracy	None
REMT	MAE	None
Choquistic regression	Accuracy, AUC	None
VC-DRSA with non-monot. features	Accuracy	None
MC-SVM	Accurary, recall, PPV, NPV, *F*-measure, κ coefficient	FOM
MGain	Accuracy	None
FREMT	Accuracy, MAE	None
MonRF	Accuracy, MAE	NMI
VC-DRSA ORF	None	None
RDMT(H)	Accuracy, κ coefficient, MAE	NMI
RMC-FSVM	Accuracy, recall, PPV, *F*-measure	None
VC-RF	Accuracy, MAE	None
MoNGEL	Accuracy, MAE	NMI
Monot. AdaBoost	Accuracy, MAE	NMI
AntMiner+, cAnt-Miner$_{PB+MC}$	Accuracy	None
EHSMC-CHC	Accuracy, MAE, MAcc, MMAE	NMI
XGBoost	AUC	None
PM-SVM	Accuracy, κ coefficient	MCC
PM-RF	Accuracy	MCC
MMT	Accuracy, MAE	None
FCMT	Accuracy, MAE	None
MCELM	MAE	None
RULEM	Accuracy, MAE, MSE	None
MFARC-HD, FS$_{MOGFS^e}$+T$_{UN^e}$	MAE, MMAE	NMI
MonoBoost	*F*-measure, κ coefficient, recall, accuracy	None
PMDT	Accuracy, MAE	None

- Monotonic Training Set Selection (*MonTSS* [101])
  MonTSS incorporates proper measurements to identify and select the most suitable instances in the training set to enhance both the accuracy and the monotonic nature of the models produced by different classifiers.

4. Quality metrics used in monotonic classification

This section analyzes and summarizes the evaluation measures used in all the experimental studies present in the specialized literature. They evaluate two different aspects: precision and monotonicity. In Table 3, we present, for each monotonic classification method, the measures used both for predictive assessment and for monotonicity fulfillment. The description of each metric is included below.

4.1. Predictive assessment metrics

In order to define the metrics considered to evaluate the predictive performance of a classifier, we introduce the following notation:

- True Positives (TP): number of instances with positive outcomes that are correctly classified.
- False Positives (FP): number of instances with positive outcomes that are incorrectly classified.
- True Negative (TN): number of instances with negative outcomes that are correctly classified.
- False Negative (FN): number of instances with negative outcomes that are incorrectly classified.

The first set of predictive measures included are applied in binary classification, and they are listed below:

- Accuracy [8]:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}, \tag{9}$$

representing the predictive ability according to the proportion of the tested data correctly classified.
- Error rate [8]:

$$\text{Error rate} = \frac{FP + FN}{TP + FP + TN + FN}. \tag{10}$$

This is the opposite case to the previous one, evaluating the proportion of the tested data incorrectly classified.
- Recall [8]:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{11}$$

Recall (also called sensitivity) is a measure of the proportion of actual positives that are correctly classified.
- Positive predictive value (PPV [8]):

$$\text{PPV} = \frac{TP}{TP + FP}, \tag{12}$$

which is the proportion of test instances with positive predictive outcomes that are correctly predicted. PPV (also known as precision) represents the probability that a positive test reflects the underlying condition being tested for.
- Negative predictive value (NPV [8]):

$$\text{NPV} = \frac{TN}{TN + FN}, \tag{13}$$

which is the proportion of test instances with negative predictive outcomes that are correctly predicted.
- *F*-measure [8]:

$$F - measure = \frac{2 \cdot PPV \cdot Recall}{PPV + Recall}. \tag{14}$$

This metric is the harmonic mean of precision and recall.
- The κ coefficient [8] represents the agreement between the classifier and the data labels, and it is computed as follows:

$$\kappa \text{ coefficient} = \frac{P_a - P_e}{1 - P_e}, \tag{15}$$

where P_e is the hypothetical probability of chance agreement and P_a is the relative observed agreement between the classifier and the data. They are computed as follows:

$$\text{P}_e = \frac{(TP + FP) \cdot (TP + FN) + (TN + FP) \cdot (TN + FN)}{(TP + TN + FP + FN)^2}, \tag{16}$$

$$\text{P}_a = \frac{TP + TN}{TP + TN + FP + FN}. \tag{17}$$

- Area Under Curve (AUC): To combine the Recall and the false positive rate ($\frac{FP}{FP+TN}$) into one single metric, we first compute the two former metrics with many different thresholds (for example 0.00, 0.01, 0.02, ..., 1.00) for the logistic regression, then plot them on a single graph, with the false positive rate values on the abscissa and the Recall values on the ordinate. The resulting curve is called ROC curve, and the metric we consider is the AUC of this curve.

The second set of predictive measures have been applied to multiclass classification problems, and are listed below:

- Mean Squared Error (MSE [65]) is calculated as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i' - y_i)^2, \tag{18}$$

where n is the number of observations in the evaluated data set, y_i' the estimated class label for observation i and y_i the true class label (both represented as integer values based on their position in the ordinal scale). It measures the average of the squares of errors.
- Mean Absolute Error (MAE [65]) is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i' - y_i|. \tag{19}$$

MAE is a measure of how close predictions are to the outcomes.
- Monotonic Accuracy (MAcc [59]), computed as standard Accuracy, but only considering those examples that completely fulfill the monotonicity constraints in the test set. In other words, non-monotonic examples do not take part in the calculation of MAcc.
- Monotonic Mean Absolute Error (MMAE [59]), calculated as standard MAE, but only considering those examples that completely fulfill the monotonicity constraints in the test set.

4.2. Monotonicity fulfillment metrics

In this case, the interest is to evaluate the rate of monotonicity provided by either the predictions obtained or the model built.

Let $\mathbf{x}$ be an example from the data set *D*. *NClash*($\mathbf{x}$) is the number of examples from *D* that do not meet the monotonicity restrictions with respect to $\mathbf{x}$, and n is the number of instances in *D*. *NMonot*($\mathbf{x}$) is the number of examples from *D* that meet the monotonicity restrictions with respect to $\mathbf{x}$.

- The Non-Monotonic Index [22,102] is defined as the number of clash-pairs divided by the total number of pairs of examples in the data set:

$$NMI = \frac{1}{n(n-1)}\sum_{\mathbf{x} \in D} NClash(\mathbf{x}) \tag{20}$$

- γ_1 [31], assessed as:

$$\gamma_1 = \frac{S_+ - S_-}{S_+ + S_-}, \tag{21}$$

$$S_- = \sum_{\mathbf{x} \in D} NClash(\mathbf{x}), \tag{22}$$

$$S_+ = \sum_{\mathbf{x} \in D} NMonot(\mathbf{x}), \tag{23}$$

where S_- is the number of discordant pairs, and S_+ is the number of concordant pairs. γ_1 is the Goodman–Kruskal's γ statistic [103].
- γ_2 [31]:

$$\gamma_2 = \frac{S_+ - S_-}{\#P}, \tag{24}$$

where $\#P$ is the total number of pairs, i.e. $P = S_+ + S_- + \#NCP$, $\#NCP$ standing for number of non-comparable pairs.
- Frequency of Monotonicity (FOM [8]):

$$\text{FOM} = \frac{S_+}{\#P}. \tag{25}$$

Table 4
Number of times each metric is used in monotonic classification literature.

Metric	# of times used
Accuracy	24
MAE	21
Error rate	5
κ coefficient	4
MSE	4
Recall	3
F-measure	3
PPV	2
MMAE	2
AUC	2
NPV	1
MAcc	1
NMI	8
MCC	2
γ_1	1
γ_2	1
FOM	1
NMI2	0

- The Non-Monotonicity Index 2 (NMI2 [104]) is defined as the number of non-monotone examples divided by the total number of examples:

$$NMI2 = \frac{1}{n}\sum_{\mathbf{x}\in D} Clash(\mathbf{x}) \tag{26}$$

where $Clash(\mathbf{x}) = 1$ if $\mathbf{x}$ clashes with at least one example in D, and 0 otherwise. If $Clash(\mathbf{x}) = 1$, $\mathbf{x}$ is called a non-monotone example. This metric was proposed in [104] but it has not been used in any study yet.
- Monotonicity Compliance (MCC [61]), defined as the proportion of the input space where the requested monotonicity constraints are not violated, weighted by the joint probability distribution of the input space. This metric has been proposed to be applied when partial monotonicity is present.

Table 4 includes the number of times each metric was used in the different experimental studies. As can be observed, the most commonly used metrics for predictive purposes are Accuracy and MAE, whereas NMI is the most popular one for estimating the monotonicity fulfillment.

5. Data sets used in monotonic classification

Next, we review monotonic classification papers to summarize which are the data sets considered in their experimental analysis.

The information about the most commonly used data sets (with at least 15 appearances in the literature) has been included in Table 5, which summarizes their properties. For each data set, we can observe the number of examples (Ex.), attributes (Atts.), numerical attributes (Num.) and nominal attributes (Nom.), the number of classes (Cl.), the source where the data set can be found, the NMI metric associated with it and finally, the number of times it has been included in experimental analysis in the literature.

A brief description is now given for each of these data sets:

- AutoMPG: the data set concerns city-cycle fuel consumption given in miles per gallon (Mpg).
- BostonHousing: the data set concerns the housing values in the suburbs of Boston.
- Car: this data set (Car Evaluation Database) was derived from a simple hierarchical decision model. The model evaluates cars according to six input attributes: buying, maint, doors, persons, lug_boot, safety.
- ERA: this data set was originally gathered during an academic decision-making experiment aiming at determining which are the most important qualities of candidates for a certain type of jobs.
- ESL: in this case, we find profiles of applicants for certain industrial jobs. Expert psychologists from a recruiting company, based on psychometric test results and interviews with the candidates, determined the values of the input attributes. The output is an overall score corresponding to the degree to which of the candidate fits this type of job.
- LEV: this data set contains examples of anonymous lecturer evaluations, taken at the end of MBA courses. Before receiving the final grades, students were asked to score their lecturers according to four attributes such as oral skills and contribution to their professional/general knowledge. The single output was a total evaluation of the lecturer's performance.
- Pima: this data set comes from the National Institute of Diabetes and Digestive and Kidney Diseases. Several constraints were placed on the selection of sample from a larger database. In particular, all patients here are females of Pima Indian heritage, and are at least 21 years old. The class label demonstrates if the person has (or not) diabetes.
- MachineCPU: this problem focuses on relative CPU performance data. The task is to approximate the published relative performance of the CPU.
- SWD: this data set contains real-world assessments of qualified social workers regarding the risk of a group of children if they stay with their families at home. This evaluation of risk assessment is often presented to judicial courts to help decide what is in the best interest of an allegedly abused or neglected child.

Considering these data sets, Table 6 includes the estimation of the possible monotonic relationship between each input feature and the class feature, by using the RMI measure [45]. This metric takes values in the range $[-1, 1]$, where -1 means that the relationship is totally inverse (if the feature increases, the class decreases), and 1 represents a completely direct relationship (if the feature increases, the class increases). If the relationship is direct (for instance, a value in the range [0.1,1]), we include a '+' in the cell. In the case of an inverse relationship (a value in the range $[-1, -0.1]$), the symbol used is '−', and, when the RMI value is in the range $[-0.1, 0.1]$, we consider that the feature and the class are not related (represented by a '='). The RMI value is given below each corresponding symbol. As can be checked in Table 6, most of the characteristics present a relationship with the corresponding class, so that they are good candidate data sets to be used in future experimental studies.

6. Guidelines and future work in monotonic classification

This section offers suggestions to researchers interested in developing new ideas within this field. We will emphasize some relevant algorithms proposed in the literature to be considered as contestant methods in experimental comparisons. In this regard, our considerations on their analysis will focus on:

- Algorithms to consider for future study: We will choose a subset of methods depending on the specific family they belong to. We will suggest a list of algorithms motivated by their properties, reputation and performance.
  - Instance-based techniques: The OSDL is a method to keep in mind due to its interpretation of the monotonicity constraints in terms of stochastic dominance which is very useful when trying to achieve total monotonicity in the predictive decisions. Furthermore, we should consider MkNN based on the basis of its simplicity, perfor-

Table 5
Summary of the most used data sets used in the monotonic classifiers literature.

Data set	Ex.	Atts.	Num.	Nom.	Cl.	Source	NMI	# of times used
AutoMPG	392	7	7	0	10	[105]	0.023	17
BostonHousing	506	12	10	2	4	[106]	0.001	15
Car	1728	6	0	6	4	[105]	0.000	22
ERA	1000	4	4	0	9	[69]	0.016	15
ESL	488	4	4	0	9	[69]	0.004	18
LEV	1000	4	4	0	5	[69]	0.006	15
MachineCPU	209	6	6	0	4	[105]	0.001	19
Pima	768	8	8	0	2	[105]	0.015	16
SWD	1000	10	10	0	4	[69]	0.009	16

Table 6
RMI measure [45] for all input features when considering the most popular monotonic classification data sets.

Data Set	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
AutoMPG	− −0.5	− −0.8	− −0.8	− −0.7	+ 0.3	+ 0.6	+ 0.4					
BostonHousing	− −0.5	+ 0.3	− −0.4	= 0.0	− −0.4	+ 0.6	− −0.4	+ 0.2	− −0.2	− −0.4	− −0.5	= 0.0
Car	+ 1.0	+ 1.0	+ 1.0	+ 1.0	+ 1.0	+ 1.0						
ERA	+ 0.3	+ 0.4	+ 0.2	+ 0.2								
ESL	+ 0.6	+ 0.6	+ 0.6	+ 0.6								
LEV	+ 0.2	+ 0.4	+ 0.2	+ 0.2								
MachineCPU	− −0.6	+ 0.6	+ 0.7	+ 0.7	+ 0.5	+ 0.5						
Pima	+ 0.2	+ 0.3	= 0.0	= 0.0	+ 0.2	+ 0.2	+ 0.2	+ 0.2				
SWD	+ 0.2	+ 0.2	+ 0.3	= 0.0	+ 0.2	= 0.0	+ 0.2	= 0.0	+ 0.2	+ 0.2		

mance and the ease of its integration when hybridized with other algorithms.

– Statistical based methods: MonMLP should be considered to be the classic multi-layer perceptron network that has been a source of inspiration for the rest of the algorithms from the same family. Another choice, which is choquistic regression, replaces the linear function of predictor variables using the choquet integral. The choquet integral is very attractive for machine learning as to if offers measures that quantify the importance of individual predictor variables and the interaction between groups of variables. We also recommend considering the PM-SVM technique because of its apability to treat partial monotonicity using an alternative metric to NMI, called MCC and its ability to measure the monotonicity degree. Finally, we should take into consideration the algorithm MCELM. Its advantages are that it does not need to tune parameters iteratively, it has extremely fast training times, does not require monotonic relationships to exist between features, the outputs are consistent and experimentally it shows generalization capability.
– Rules and Decision Trees family: MID was the first proposal in this family. The idea is simple and intuitive since it consists of the inclusion of a criterion to achieve a trade-off between accuracy and the monotonicity constraints present in the data. In fact, this criterion can be easily attached to any decision tree and rule learning model. Also noteworthy is the fact that its performance in prediction is oustanding. There is another decision tree algorithm called REMT which introduces the rank mutual information (RMI) as a feature quality measure, combining the advantage of robustness of Shannons entropy with the ability of dominance rough sets in extracting ordinal structures from monotonic data sets. The models generated by REMT are monotonically consistent and have high predictive capabilities. Finally, we highlight the choice of MMT based on its ability to handle the performance limitations produced by incomparable object pairs. MMT addresses this problem constructing multivariate decision trees with monotonicity constraints.
– Ensemble based methods: Five techniques have been selected as the most noteworthy. The LPRules and MonoBoost are representative boosting algorithms for multiclass problems, despite the first was based on binary decomposition. FREMT has been chosen thanks to the interesting attribute reduction and fusing principle introduced in its definition. MonRF is considered due to its good performance and the extensive experimental analysis conducted by the authors. Lastly, the scalability of XGBoost and its capability of obtaining monotone consistent decisions makes it an obvious choice.

• Quality metrics: the quality of the models learned can be evaluated based on precision or monotonicity fulfillment. If we take precision into account, MAE is the measure that should be considered as it is widely used in the area. As far as monotonicity fulfillment is concerned, NMI is the most commonly used metric which adequately reflects compliance with model monotonicity (see Section 4.1).

Directions for future research in monotonic classification are indicated as follows:

• It is necessary to propose performance measures that combine the evaluation of accurate and monotonic predictions. Currently, MAE and NMI measures are mainly used simultaneously, but the latter requires a complete set to calculate the comparability of the examples. We have seen that in some revised algorithms useful measures have been proposed for this purpose, but they are hardly being used in successive ideas. On the other hand, we are missing the use of complex measures that have been used in ordinal regression [107], such as ROC curves or performance curves.

- The distinction between partial and total monotonic classification is crucial and this should be clearly indicated in future proposals. Depending on the application, it will make more sense to use one type of technique or another, depending on where the importance of the model learned lies; either in the interpretation of the model or in the accuracy of the model. We recommend that this type of property be clearly highlighted in future proposals.
- It is possible to devise extensions of the classic monotonic classification problem based on different graduations of restrictions between input and output attributes. There may be attributes that are more relevant than others in the monotonicity constraint and their violation may result in greater perjury. This implies a reformulation of the partial order and a generalization of the problem to introduce bias in the predictions.
- Although adaptations of all types of classifiers to this problem, including ensembles, have been proposed, other types of proposals are still lacking, such as the decomposition of One-Versus-One (OVO) and more advanced One-Versus-All (OVA) classes [5,108] and more data preprocessing techniques, such as noise filtering [109].
- We have also observed that many of the algorithms reviewed in this paper are not available to the public in software repositories. More software development is needed in this area.
- Currently, monotonic classification is understood as a natural extension of the classical or ordinal regression. Other predictive learning paradigms that require some interpretation of the results may benefit from monotone models or monotone predictions in certain real-life applications. We refer to those singular or non-standard predictive problems [110] including weak supervision [111]. To date, there are proposals to deal with monotonicity constraints in imbalanced classification [112,113].

7. Conclusions

This paper is a systematical review of monotonic classification literature that could be used as a functional guide on the scope. Monotonic classification is an emerging area in the field of data mining. In recent years, the number of proposals in this area of knowledge has significantly increased, as shown in Fig. 1. This fact justifies the necessity of proposing a taxonomy that classifies and discriminates all the methods proposed so far. The taxonomy designed can be used as a guide to:

- Decide which kind of algorithm and model is best suited for a new monotonic problem.
- Compare any new proposals with those current proposals which come from the same family, so that it can be decided if the new proposal should be considered and if any improvements in their performance can be observed.

Together with this taxonomy, we also analyze which methods are publicly available, and whose source codes are available on line. In those cases, we also include where their implementation can be found.

Additionally, an analysis of the proposed and used quality metrics is carried out, considering predictive assessment and monotonicity fulfillment. We also highlight some measures, which are more frequently considered in this field, such as Accuracy, MAE and NMI.

Finally, a summary and description of all the data sets used is considered. We emphasize eight of them, which have been used in, at least, ten of the experimental evaluations reviewed in the literature. Their characteristics, availability and the monotonic relationships between input features and the class label are also detailed.

The overview is completed by including a set of guidelines regarding the most representative methods found in the literature to be considered in novel ideas and proposals and with an enumeration of possible directions for future research in this field.

Acknowledgment

This work has been supported by TIN2017-89517-P, TIN2015-70308-REDT, TIN2014-54583-C2-1-R and the Spanish “Ministerio de Economía y Competitividad” and by “Fondo Europeo de Desarrollo Regional” (FEDER) under Project TEC2015-69496-R.

References

[1] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Series in Data Management Systems, fourth ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2016.
[2] A.I. Saleh, F.M. Talaat, L.M. Labib, A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers, Artif. Intell. Rev. (2017) 1–41.
[3] B.A. Tama, K.-H. Rhee, Tree-based classifier ensembles for early detection method of diabetes: an exploratory study, Artif. Intell. Rev. (2017) 1–16.
[4] P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernandez-Navarro, C. Hervás-Martínez, Ordinal regression methods: survey and experimental study, IEEE Trans. Knowl. Data Eng. 28 (1) (2016) 127–146, doi:10.1109/TKDE.2015.2457911.
[5] W. Kotłowski, R. Słowiński, On nonparametric ordinal classification with monotonicity constraints., IEEE Trans. Knowl. Data Eng. 25 (11) (2013) 2576–2589.
[6] J.-R. Cano, N.R. Aljohani, R.A. Abbasi, J.S. Alowidbi, S. García, Prototype selection to improve monotonic nearest neighbor, Eng. Appl. Artif. Intell. 60 (2017) 128–135.
[7] M.-J. Kim, I. Han, The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms, Exp. Syst. Appl. 25 (4) (2003) 637–646.
[8] C.-C. Chen, S.-T. Li, Credit rating with a monotonicity-constrained support vector machine model, Exp. Syst. Appl. 41 (16) (2014) 7235–7247.
[9] R. Potharst, A.J. Feelders, Classification trees for problems with monotonicity constraints, SIGKDD Explor. 4 (1) (2002) 1–10.
[10] A. Ben-David, Automatic generation of symbolic multiattribute ordinal knowledge-based DSSs: methodology and applications, Decis. Sci. 23 (1992) 1357–1372.
[11] P.A. Gutiérrez, S. García, Current prospects on ordinal and monotonic classification, Progr. Artif. Intell. 5 (3) (2016) 171–179.
[12] H. Zhu, E.C. Tsang, X.-Z. Wang, R.A.R. Ashfaq, Monotonic classification extreme learning machine, Neurocomputing 225 (2017) 205–213.
[13] J.S. Cardoso, R. Sousa, Measuring the performance of ordinal classification, Int. J. Pattern Recognit. Artif. Intell. 25 (8) (2011) 1173–1195.
[14] W. Kotłowski, The Paradox of Overfitting, Poznan University of Technology, 2008 Master's thesis.
[15] C.C. Aggarwal, Data Mining: The Textbook, Springer, 2015.
[16] S. García, J. Luengo, F. Herrera, Data Preprocessing in Data Mining, Springer, 2015.
[17] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1) (2010) 1–39.
[18] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[19] A.F. Tehrani, W. Cheng, K. Dembczyński, E. Hüllermeier, Learning monotone nonlinear models using the Choquet integral, Mach. Learn. 89 (1–2) (2012) 183–211.
[20] H. Daniels, M. Velikova, Monotone and partially monotone neural networks, IEEE Trans. Neural Netw. 21 (6) (2010) 906–917.
[21] E. Frank, M. Hall, I. Witten, The Weka workbench, Data Min. Pract. Mach. Learn. Tools Tech. 4 (2016). https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.
[22] A. Ben-David, Monotonicity maintenance in information-theoretic machine learning algorithms, Mach. Learn. 19 (1) (1995) 29–43.
[23] M. Grabisch, A new algorithm for identifying fuzzy measures and its application to pattern recognition, in: Proceedings of the 1995 IEEE International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium, 1, IEEE, 1995, pp. 145–150.
[24] J. Sill, Monotonic networks, in: Proceedings of the 1997 Advances in Neural Information Processing Systems, 1997, pp. 661–667.

[25] M. Kazuhisa, S. Takashi, O. Hirotaka, I. Toshihide, Data analysis by positive decision trees, IEICE Trans. Inf. Syst. E82-D (1) (1999) 76–88.
[26] R. Dykstra, J. Hewett, T. Robertson, Nonparametric, isotonic discriminant procedures, Biometrika 86 (2) (1999) 429–438.
[27] R. Potharst, J.C. Bioch, Decision trees for ordinal classification, Intell. Data Anal. 4 (2) (2000) 97–111.
[28] S. Greco, B. Matarazzo, R. Slowinski, J. Stefanowski, Variable consistency model of dominance-based rough sets approach, in: Proceedings of the International Conference on Rough Sets and Current Trends in Computing, Springer, 2000a, pp. 170–181.
[29] S. Greco, B. Matarazzo, R. Slowinski, J. Stefanowski, An algorithm for induction of decision rules consistent with the dominance principle, in: Proceedings of the International Conference on Rough Sets and Current Trends in Computing, Springer, 2000b, pp. 304–313.
[30] J. Bioch, V. Popova, Monotone decision trees and noisy data, in: Proceedings of the Fourteenth Belgium–Dutch Conference on Artificial Intelligence, 2002, pp. 19–26.
[31] J.W. Lee, D.S. Yeung, X. Wang, Monotonic decision tree for ordinal classification, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 3, IEEE, 2003, pp. 2623–2628.
[32] R. Chandrasekaran, Y.U. Ryu, V.S. Jacob, S. Hong, Isotonic separation, INFORMS J. Comput. 17 (4) (2005) 462–474.
[33] B. Lang, Monotonic multi-layer perceptron networks as universal approximators, Proceedings of the International Conference on Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, 2005. 750–750.
[34] J. Błaszczyński, S. Greco, R. Słowiński, Multi-criteria classification – a new scheme for application of dominance-based decision rules, Eur. J. Oper. Res. 181 (3) (2007) 1030–1044.
[35] S. Lievens, B.D. Baets, K. Cao-Van, A probabilistic framework for the design of instance-based supervised ranking algorithms in an ordinal setting, Ann. Oper. Res. 163 (1) (2008) 115–142.
[36] W. Duivesteijn, A. Feelders, Nearest neighbour classification with monotonicity constraints, in: Proceedings of the 2008 ECML/PKDD, in: Lecture Notes in Computer Science, 5211, Springer, 2008, pp. 301–316.
[37] N. Barile, A. Feelders, Nonparametric monotone classification with MOCA, in: Proceedings of the Eighth IEEE International Conference on Data Mining, ICDM'08, IEEE, 2008, pp. 731–736.
[38] W. Kotłowski, K. Dembczyński, S. Greco, R. Słowiński, Stochastic dominance-based rough set model for ordinal classification, Inf. Sci. 178 (21) (2008) 4019–4037.
[39] R. Van De Kamp, A. Feelders, N. Barile, N. Adams, Isotonic classification trees, in: Proceedings of the International Symposium on Intelligent Data Analysis, IDA 2009, Springer, 2009, pp. 405–416.
[40] W. Kotłowski, R. Słowiński, Rule learning with monotonicity constraints, in: Proceedings of the Twenty-Sixth Annual International Conference on Machine Learning, ACM, 2009, pp. 537–544.
[41] M. Inuiguchi, Y. Yoshioka, Y. Kusunoki, Variable-precision dominance-based rough set approach and attribute reduction, Int. J. Approx. Reason. 50 (8) (2009) 1199–1214.
[42] K. Dembczyński, W. Kotłowski, R. Słowiński, Learning rule ensembles for ordinal classification with monotonicity constraints, Fundam. Inf. 94 (2) (2009) 163–178.
[43] J. Błaszczyński, R. Słowiński, J. Stefanowski, Rough sets and current trends in computing, in: Proceedings of the Seventh International Conference on Rough Sets and Current Trends in Computing, RSCTC 2010, Warsaw, Poland, June 28–30, 2010, Springer, Berlin, Heidelberg, pp. 392–401.
[44] J. Blaszczynski, R. Slowinski, M. Szelkag, Sequential covering rule induction algorithm for variable consistency rough set approaches, Inf. Sci. 181 (5) (2011) 987–1002.
[45] Q. Hu, X. Che, L. Zhang, D. Zhang, M. Guo, D. Yu, Rank entropy-based decision trees for monotonic classification, IEEE Trans. Knowl. Eng. 24 (11) (2012) 2052–2064.
[46] J. Blaszczynski, S. Greco, R. Slowinski, Inductive discovery of laws using monotonic rules, Eng. Appl. Artif. Intell. 25 (2) (2012) 284–294.
[47] J. Zhang, J. Zhai, H. Zhu, X. Wang, Induction of monotonic decision trees, in: Proceedings of the 2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), IEEE, 2015, pp. 203–207.
[48] Y. Qian, H. Xu, J. Liang, B. Liu, J. Wang, Fusing monotonic decision trees, IEEE Trans. Knowl. Data Eng. 27 (10) (2015) 2717–2728.
[49] S. González, F. Herrera, S. García, Monotonic random forest with an ensemble pruning mechanism based on the degree of monotonicity, New Gen. Comput. 33 (4) (2015) 367–388.
[50] S. Wang, J. Zhai, S. Zhang, H. Zhu, An ordinal random forest and its parallel implementation with MapReduce, in: Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2015, pp. 2170–2173.
[51] J. Błaszczyński, S. Greco, B. Matarazzo, R. Słowiński, M. Szelag, jMAF-dominance-based rough set data analysis framework, Rough Sets and Intelligent Systems – Professor Zdzisław Pawlak in Memoriam, Springer, 2013, pp. 185–209. http://www.cs.put.poznan.pl/jblaszczynski/Site/jRS.html.
[52] C. Marsala, D. Petturiti, Rank discrimination measures for enforcing monotonicity in decision tree induction, Inf. Sci. 291 (2015) 143–171.
[53] S.-T. Li, C.-C. Chen, A regularized monotonic fuzzy support vector machine model for data mining with prior knowledge, IEEE Trans. Fuzzy Syst. 23 (5) (2015) 1713–1727.
[54] H. Wang, M. Zhou, K. She, Induction of ordinal classification rules from decision tables with unknown monotonicity, Eur. J. Oper. Res. 242 (1) (2015) 172–181.
[55] J. García, H.M. Fardoun, D.M. Alghazzawi, J.-R. Cano, S. García, MoNGEL: monotonic nested generalized exemplar learning, Pattern Anal. Appl. 20 (2) (2017) 441–452.
[56] J. García, H.M. Fardoun, D.M. Alghazzawi, J.-R. Cano, S. García, MoNGEL Java Code, 2015. http://www4.ujaen.es/~jrcano/Research/MoNGEL/sourcecode.html.
[57] S. González, F. Herrera, S. García, Managing monotonicity in classification by a pruned AdaBoost, in: Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, Springer, 2016, pp. 512–523.
[58] J. Brookhouse, F.E.B. Otero, Monotonicity in ant colony classification algorithms, in: Proceedings of the Tenth International Conference on Swarm Intelligence, ANTS 2016, Springer International Publishing, Brussels, Belgium, 2016, pp. 137–148. September 7–9, 2016.
[59] J. García, A.M. AlBar, N.R. Aljohani, J.-R. Cano, S. García, Hyperrectangles selection for monotonic classification by using evolutionary algorithms, Int. J. Comput. Intell. Syst. 9 (1) (2016) 184–201.
[60] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of the Twenty-Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794. https://github.com/dmlc/xgboost/.
[61] C. Bartley, W. Liu, M. Reynolds, Effective monotone knowledge integration in kernel support vector machines, in: Proceedings of the Twelfth International Conference on Advanced Data Mining and Applications, Springer, 2016a, pp. 3–18. https://github.com/chriswbartley/PMSVM.
[62] C. Bartley, W. Liu, M. Reynolds, A novel technique for integrating monotone domain knowledge into the random forest classifier, in: Proceedings of the Fourteenth Australasian Data Mining Conference, 170, 2016b, pp. 3–18. https://github.com/chriswbartley/PMRF.
[63] S. Pei, Q. Hu, C. Chen, Multivariate decision trees with monotonicity constraints, Knowl.-Based Syst. 112 (2016) 14–25.
[64] H. Xu, W. Wang, Y. Qian, Fusing complete monotonic decision trees, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2223–2235.
[65] W. Verbeke, D. Martens, B. Baesens, RULEM: a novel heuristic rule learning approach for ordinal classification with monotonicity constraints, Appl. Soft Comput. (2017), doi:10.1016/j.asoc.2017.01.042.
[66] J. Alcalá-Fdez, R. Alcala, S. González, Y. Nojima, S. Garcia, Evolutionary fuzzy rule-based methods for monotonic classification, IEEE Trans. Fuzzy Syst. 25 (6) (2017) 1376–1390.
[67] C. Bartley, W. Liu, M. Reynolds, A novel framework for constructing partially monotone rule ensembles, in: Proceedings of the Thirty-Fourth International Conference on Data Engineering, 2018, pp. 1320–1323. https://github.com/chriswbartley/monoboost.
[68] S. Pei, Q. Hu, Partially monotonic decision trees, Inf. Sci. 424 (2018) 104–117.
[69] A. Ben-David, L. Sterling, Y.H. Pao, Learning, classification of monotonic ordinal concepts, Comput. Intell. 5 (1989) 45–49.
[70] S. Lievens, B. De Baets, Supervised ranking in the Weka environment, Inf. Sci. 180 (24) (2010) 4763–4771.
[71] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[72] J.R. Quinlan, C4.5: Programs for Machine Learning, Elsevier, 2014.
[73] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[74] I. Triguero, D. Peralta, J. Bacardit, S. García, F. Herrera, MRPR: a mapreduce solution for prototype reduction in big data classification, Neurocomputing 150 (2015) 331–345.
[75] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the European Conference on Computational Learning Theory, Springer, 1995, pp. 23–37.
[76] M. Seligman, Rborist: Extensible, Parallelizable Implementation of the Random Forest Algorithm, 2017. https://cran.r-project.org/web/packages/Rborist/index.html
[77] B. Greenwell, B. Boehmke, J. Cunningham, G. Developers, GBM: Generalized Boosted Regression Models, 2018. https://cran.r-project.org/web/packages/gbm/index.html.
[78] K. Pelckmans, M. Espinoza, J. De Brabanter, J.A. Suykens, B. De Moor, Primal–dual monotone kernel regression, Neural Process. Lett. 22 (2) (2005) 171–182.
[79] A.N. Tikhonov, V. Arsenin, Solutions of Ill-Posed Problems, 14, Winston Washington, DC, 1977.
[80] M. Grabisch, J.-M. Nicolas, Classification by fuzzy integral: Performance and tests, Fuzzy Sets Syst. 65 (2–3) (1994) 255–271.
[81] M. Grabisch, Modelling data by the Choquet integral, in: Information Fusion in Data Mining, Springer, 2003, pp. 135–148.
[82] A.F. Tehrani, E. Hüllermeier, Ordinal choquistic regression, in: Proceedings of the 2013 Conference on EUSFLAT, 2013, pp. 1–8.
[83] M. Grabisch, Fuzzy integral in multicriteria decision making, Fuzzy Sets Syst. 69 (3) (1995) 279–298.
[84] H. Daniels, M. Velikova, et al., Derivation of Monotone Decision Models from Non-Monotone Data, Tilburg University, 2003.
[85] A. Feelders, M. Velikova, H. Daniels, Two polynomial algorithms for relabeling non-monotone data, Technical Report, 2006.
[86] M. Rademaker, B. De Baets, H. De Meyer, Loss optimal monotone relabeling of noisy multi-criteria data sets, Inf. Sci. 179 (24) (2009) 4089–4096.
[87] A. Feelders, Monotone relabeling in ordinal classification, in: Proceedings of the Tenth IEEE International Conference on Data Mining (ICDM), IEEE, 2010, pp. 803–808.

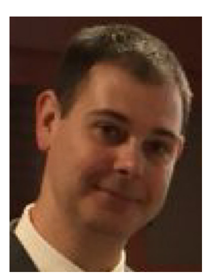
[88] L. Stegeman, A. Feelders, On generating all optimal monotone classifications, in: Proceedings of the Eleventh IEEE International Conference on Data Mining (ICDM), IEEE, 2011, pp. 685–694.
[89] A. Feelders, T. Kolkman, Exploiting monotonicity constraints to reduce label noise: an experimental evaluation, in: Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 2148–2155.
[90] M. Rademaker, B. De Baets, H. De Meyer, Optimal monotone relabelling of partially non-monotone ordinal data, Optim. Methods Softw. 27 (1) (2012) 17–31.
[91] W. Pijls, R. Potharst, Repairing Non-Monotone Ordinal Data Sets by Changing Class Labels, Technical Report, Econometric Institute, Erasmus University Rotterdam, 2014.
[92] S. Kotsiantis, Feature selection for machine learning classification problems: a recent overview, Artif. Intell. Rev. (2011) 1–20.
[93] Q. Hu, W. Pan, Y. Song, D. Yu, Large-margin feature selection for monotonic classification, Knowl.-Based Syst. 31 (2012a) 8–18.
[94] Q. Hu, W. Pan, L. Zhang, D. Zhang, Y. Song, M. Guo, D. Yu, Feature selection for monotonic classification, IEEE Trans. Fuzzy Syst. 20 (1) (2012b) 69–81.
[95] W. Pan, Q. Hu, Y. Song, D. Yu, Feature selection for monotonic classification via maximizing monotonic dependency, Int. J. Comput. Intell. Syst. 7 (3) (2014) 543–555.
[96] W. Pan, Q. Hu, An improved feature selection algorithm for ordinal classification, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 99 (12) (2016) 2266–2274.
[97] H. Yang, Z. Xu, M.R. Lyu, I. King, Budget constrained non-monotonic feature selection, Neural Netw. 71 (2015) 214–224.
[98] J.R. Cano, F. Herrera, M. Lozano, Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study, IEEE Trans. Evol. Comput. 7 (6) (2003) 561–575.
[99] S. Garcia, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435.
[100] J.R. Cano, F. Herrera, M. Lozano, Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability, Data Knowl. Eng. 60 (1) (2007) 90–108.
[101] J.-R. Cano, S. García, Training set selection for monotonic ordinal classification, Data Knowl. Eng. 112 (2017) 94–105.
[102] H. Daniels, M. Velikova, Derivation of monotone decision models from noisy data, IEEE Trans. Syst. Man Cybern. Part C 36 (2006) 705–710.
[103] L. Goodman, W. Kruskal, Measures of Association for Cross Classifications, Springer-Verlag, 1977.
[104] I. Milstein, A. Ben-David, R. Potharst, Generating noisy monotone ordinal datasets, Artif. Intell. Res. 3 (1) (2014) 30–37.
[105] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Mult.-Valued Logic Soft Comput. 17 (2–3) (2011) 255–287.
[106] K. Bache, M. Lichman, UCI Machine Learning Repository, 2013, https://archive.ics.uci.edu/ml/index.php.
[107] W. Waegeman, B. De Baets, L. Boullart, ROC analysis in ordinal regression learning, Pattern Recognit. Lett. 29 (1) (2008) 1–9.
[108] Z. Zhang, X. Luo, S. González, S. García, F. Herrera, DRCW-ASEG: one-versus-one distance-based relative competence weighting with adaptive synthetic example generation for multi-class imbalanced datasets, Neurocomputing 285 (2018) 176–187.
[109] J.R. Cano, J. Luengo, S. García, Label noise filtering techniques to improve monotonic classification, Neurocomputing (2019) in press, doi:10.1016/j.neucom.2018.05.131.
[110] D. Charte, F. Charte, S. García, F. Herrera, A snapshot on nonstandard supervised learning problems: taxonomy, relationships, problem transformations and algorithm adaptations, Progr. Artif. Intell. (2019) in press, doi:10.1007/s13748-018-00167-7.
[111] J. Hernández-González, I. Inza, J.A. Lozano, Weak supervision and other non-standard classification problems: a taxonomy, Pattern Recognit. Lett. 69 (2016) 49–55.
[112] A. Fernández, S. García, M. Galar, R.C. Prati, B. Krawczyk, F. Herrera, Learning from Imbalanced Data Sets, Springer, 2018.
[113] S. González, S. García, S. Li, F. Herrera, Chain based sampling for monotonic imbalanced classification, Inf. Sci. 474 (2019) 187–204.

José Ramón Cano received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1999 and 2004, respectively. He is currently a Professor in the Department of Computer Science, University of Jaén, Jaén, Spain. His research interests include data mining, data reduction, data complexity, interpretability-accuracy trade-off, motonic classification and evolutionary algorithms.

Pedro Antonio Gutiérrez 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 with the Department of Computer Science and Numerical Analysis, University of Córdoba (Spain). His research interests are in the areas of supervised learning, evolutionary artificial neural networks, ordinal classification and the application of these techniques to different real world problems, including precision agriculture, renewable energy, climatology and biomedicine, among others.

Bartosz Krawczyk is an assistant professor in the Department of Computer Science, Virginia Commonwealth University, Richmond VA, USA, where he heads the Machine Learning and Stream Mining Lab. He obtained his M.Sc. and Ph.D. degrees from Wroclaw University of Science and Technology, Wroclaw, Poland, in 2012 and 2015 respectively. His research is focused on machine learning, data streams, ensemble learning, class imbalance, one-class classifiers, and interdisciplinary applications of these methods. He has authored 45+ international journal papers and 100+ contributions to conferences. He was awarded with numerous prestigious awards for his scientific achievements like IEEE Richard Merwin Scholarship and IEEE Outstanding Leadership Award among others. He served as a Guest Editor in four journal special issues and as a chair of ten special session and workshops. He is a member of Program Committee for over 40 international conferences and a reviewer for 30 journals.

Michał Woźniak is a professor of computer science at the Department of Systems and Computer Networks, Wrocław University of Science and Technology, Poland. He received M.Sc. degree in biomedical engineering from the Wrocław University of Technology in 1992, and Ph.D. and D.Sc. (habilitation) degrees in computer science in 1996 and 2007, respectively, from the same university. In 2015 he was nominated as the professor by President of Poland. His research focuses on machine learning, compound classification methods, classifier ensembles, data stream mining, and imbalanced data processing. He has been involved in research projects related to the above-mentioned topics and has been a consultant of several commercial projects for well-known Polish companies and public administration. He has published over 260 papers and three books. His recent one Hybrid classifiers: Method of Data, Knowledge, and Data Hybridization was published by Springer in 2014. He was awarded with numerous prestigious awards for his scientific achievements as IBM Smarter Planet Faculty Innovation Award (twice) or IEEE Outstanding Leadership Award, and several best paper awards of the prestigious conferences. He serves as program committee chairs and member for the numerous scientific events and prepared several special issues as the guest editor. He is the member of the editorial board of the high ranked journals as Information Fusion (Elsevier), Applied Soft Computing (Elsevier), and Engineering Applications of Artificial Intelligence (Elsevier). He is a senior member of the IEEE.



Salvador García is currently an Associate Professor in the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. He has published more than 80 papers in international journals (more than 60 in Q1), with more than 6000 citations, h-index 41, over 60 papers in international conference proceedings (data from Web of Science). He has been associated with the international program committees and organizing committees of several regular international conferences including IEEE CEC, ICPR, ICDM, IJCAI, etc. As edited activities, he has co-edited two special issues in international journals and he is an associate editor of "Information Fusion" (Elsevier), "Swarm and Evolutionary Computation" (Elsevier) and "AI Communications" (IOS Press) journals, and he is co-Editor in Chief of the international journal "Progress in Artificial Intelligence" (Springer). He is a co-author of the books entitled "Data Preprocessing in Data Mining" and "Learning from Imbalanced Data Sets" published by Springer. His research interests include data science, data preprocessing, Big Data, evolutionary learning, Deep Learning, metaheuristics and biometrics. He has been given some awards and honors for his personal work or for his publications in and conferences, such as IFSA-EUSFLAT 2015 Best Application Paper Award and IDEAL 2015 Best Paper Award. He belongs to the list of the Highly Cited Researchers in the area of Computer Sciences (2014–2018): http://highlycited.com/ (Clarivate Analytics). His h-index is 41 in Scholar Google, receiving more than 13,000 citations, till date.