Data Driven Transformation of a Classification Model into Ranking

Salem Chakhar
Portsmouth Business School
University of Portsmouth
Portsmouth, UK
salem.chakhar@port.ac.uk

Yu-ling Lin
School of Management
University of Bath
Bath, UK
yll41@bath.ac.uk

Rui Yang
Portsmouth Business School
University of Portsmouth
Portsmouth, UK
rui.yang@port.ac.uk

***Abstract*—Given a set of decision objects, an ordinal classifier is an algorithm that can group these objects into preference-ordered decision classes. A ranker is an algorithm that can sort a set of decision objects from highest to lowest, typically using a scoring function. In this paper we propose a ranker to transform the output of ordinal classifier into a ranking using elementary preference information extracted from the dataset. The proposed approach is illustrated using green car data in Taiwan.**

***Index Terms*—Classification, Ranking, Preference, Scoring**

I. Introduction

A. Ordinal Classification

Classification is a common real-life problem that can be addressed using different machine learning techniques, particularly classification methods such as k-Nearest Neighbors, Decision Trees Naive Bayes and Random Forest. Most of machine learning methods for classification problems commonly assume that the classes are unordered. However, in many practical applications there is (natural or artificial) preference order on classes. The standard approach to ordinal classification converts the class value into a numeric quantity and applies a regression learner to the transformed data, translating the output back into a discrete class value in a post-processing step [2]. However, this method can only be applied in conjunction with a regression scheme.

Machine learning literature contains some new or extended classification algorithms that use ordering information in class attributes, e.g. [1] [2] [5]. However, most of these methods consider only the natural order on attributes values and fail to support user preferences with respect to attributes' values. Closely related to machine learning methods, there are a range of multicriteria ordinal classification (often refereed to sorting) special-purpose algorithms that are specifically devoted to deal with classification problems involving preference ordered attributes and classes. Multicriteria ordinal classification methods have the same working principle as classical machine learning methods for classification problems, with additional features enabling them to explicitly take into account preference-ordered attributes and classes. The most known multicriteria ordinal classification are UTA, ELECTRE TRI and DRSA [4] [5] [8].

B. Motivations

Grouping objects into preference ordered classes provides some general insights on these objects. This maybe enough in several problems where the objective is to assign a class label to each object. However, in several real-life problems we need to go beyond grouping objets into classes to rank order these objects. In a recruitment problem, for example, a binary classification can be used to group candidates into those that should be interviewed and those that should be rejected. For those shortlisted, they need to be ranked order so a final decision can be taken.

A more complex problem is spare parts management where typically spare parts are grouped into three preference classes, namely A, B and C [6]. Spare parts classified in group A need to be tightly management and reorder points need to be specified for each of them, while a few of them should be held in inventory and ordered frequently. Reorder points should also be defined for spare parts classified in group B, but the management of this group needs less attention than those in group A–only periodic review is needed here. Finally, spare parts classified in group C should be kept in stock and ordered when required. However, assigning spare parts into A, B an C classes does not provide an optimised inventory system for each group of spare parts. When the number of spare parts is very large, further analysis is needed in order to identify the most critical spare parts. A possible solution to this problem is to use advanced optimisation techniques such as genetic programming [3] or joint optimization [10]. Another solution consists in applying a ranking algorithm to rank order spare parts in group A according to their criticality.

Naturally, it is possible to apply a ranking model on the dataset, without passing through a classification step. This strategy can be justified for a reduced dataset, but it will be highly demanding in computing time for large datsets. This is because most of ranking models require to pairwise the objects while a classification will only consider intrinsic characteristics of objects. Hence, the strategy advocated in this paper is to go through a preliminary classification step followed by a ranking step, applied on all or a subset of classes.

C. Contribution

In this paper, we assume that the objects have been grouped into a set of predefined and preference decision classes. The objective is then to rank order them while respecting the preference ordered established by the classes. To this end, we propose a score-based ranking algorithm permitting to transform the output of ordinal classification into a ranking while using elementary preference information extracted from the dataset and respecting the preference ordered established by the classes. This means that objects from a given class souled be ranked higher to any other object from less preferred classes. The proposed approach is illustrated using green car data in Taiwan.

The paper is organised as follows. Section II introduces the problem considered. Section III details the scoring function. Section IV introduces the score-based ranking algorithm. Section V applies the proposed approach to green car model. Section VI concludes the paper.

II. Problem Description

Let U be a non-empty finite set of decision objects described over a non-empty finite set of attributes Q. The set Q is often divided into a sub-set $C \neq \emptyset$ of *condition attributes* and a sub-set $D \neq \emptyset$ of *decision attributes*, such that $C \cup D = Q$ and $C \cap D = \emptyset$. The evaluation of an object $x \in U$ with respect to condition attribute $q \in C$ is denoted by $f(x, q)$. The domains of attributes are supposed to be ordered according to a decreasing or increasing preference. Three types of condition attributes are considered:

- gain-type: the more, the better: if the preference is increasing with the attribute values.
- cost-type: the less, the better: if the preference is decreasing with the attribute values.
- neutral: when the two first cases are not relevant.

We also assume that the set of decision attributes $D = \{d\}$ is a singleton, which is always of gain-type. The unique decision attribute d makes a partition of U into a finite number of preference-ordered decision classes $Cl_1, \ldots, Cl_n$, such that each $x \in U$ belongs to one and only one class.

The assignment of decision objects to decision classes relies on a preference learning method. The Dominance-based Rough Set Approach (DRSA) [4] [8] [9] is a typical preference learning method that is assumed to be used here. The DRSA uses a subset $L \subseteq U$ as learning set to extract prediction patterns, which typically take the form of decision rules $E \rightarrow H$ where E is the premise and H is the decision. A collection of decision rules obtained from the same dataset is called classifier [7]. The latter is then used to predict the classes of unseen objects.

A ranking is a total order, possibly with ties. The latter are represented by an equivalence relation over U, so the total order is on those equivalence classes. For notational convenience, we define a ranking function r as $r : U \times U \rightarrow \{>, =, <\}$, deciding for any pair of objects whether the first is more likely $(>)$, equally likely $(=)$, or less likely $(<)$ to be more preferred than the second.

III. Scoring Function

Let $Cl = \{Cl_t, t \in T\}$, $T = \{1, \ldots, n\}$ be an ordered set of decision classes with the following preference order:

$$Cl_1 \prec \ldots \prec Cl_t \prec \ldots \prec Cl_n$$

We denote by $Cl(x)$ the decision class to which decision object $x \in U$ is assigned. The objective of this section is to define a scoring function that assigns a positive number to each decision object in U based solely on the data available in the learning dataset. The elementary information upon which the scoring function is built are extracted by the applying the dominance relation on the learning dataset.

A. Dominance Relation

A *dominance relation* Δ associated with attributes in C is defined for each pair of objects x and y as follows:

$$x \Delta y \Leftrightarrow f(x, q) \succeq f(y, q), \forall q \in C. \quad (1)$$

In this definition, the symbol "$\geq$" should be replaced with "$\leq$" for criteria which are ordered according to decreasing preference and by "=" for those no preference order (as for example for nominal attributes).

To each object $x \in U$, we associate two sets:

- $\Delta^+(x) = \{y \in U : y \Delta x\}$.
- $\Delta^-(x) = \{y \in U : x \Delta y\}$.

The *dominating set* $\Delta^+(x)$ contains the objects that dominate x while the *dominated set* $\Delta^-(x)$ contains the objects dominated by x.

B. Weighting Systems

One strong constraint of the scoring function is that the score of decision object x must be strictly higher to the score of any decision object y for all $x, y \in U$ and $Cl(x) \succ Cl(y)$. A possible solution to ensure that this constraint is respected consists in assigning an *absolute weight* π_t to each decision class $Cl_t \in Cl$ using the following weighting system:

$$\begin{cases} \pi_1 \geq 1, \\ \pi_t = \pi_{t-1} + 2, \quad \forall\ t = 2, \ldots, n. \end{cases} \quad (2)$$

This weighting system ensures that $\pi_{cl(x)} > \pi_{cl(y)}$, $\forall\ x, y \in U$, and $Cl(x) \succ Cl(y)$.

We define also a set of *relative weights* $w^t_{t'}$ for each class $Cl_t \in Cl$ with respect to each class $Cl_{t'} \in Cl$ $(t, t' \in T)$:

$$\begin{cases} w^t_t = 1, \\ w^t_i = \lambda \cdot\ w^t_{i-1} + 2, \quad \forall\ i = t+1, \ldots, n, \\ w^t_j = \lambda \cdot\ w^t_{j+1} + 2, \quad \forall\ j = 1, \ldots, t-1. \end{cases} \quad (3)$$

The first line in Equation (3) initializes the weights and ensures that the weights are positive numbers. The second line in Equation (3) defines the weights for the decision classes $Cl_{t+1}, \ldots, Cl_n$ with a preference order higher than the decision class Cl_t under consideration. The third line in Equation (3) defines the weights for the decision classes

$Cl_1, \ldots, Cl_{t-1}$ with a preference order less than the decision class Cl_t under consideration. The multiplying factor λ in the second and third lines ensures that: (i) $w^t_{i+1} > w^t_i$ for all $i = t+1, \cdots, n$ and (ii) $w^t_j < w^t_{j+1}$ for all $j = 1, \ldots, t-1$.

C. Partial Scores

Decision objects are first discriminated based on the absolute weights π_t of their corresponding classes. It is clear that objects belonging to the same class will have exactly the same absolute weight. To better discriminate decision objects, we associate to each decision object x two partial scores. The first partial score relies on dominating $\Delta^+(\cdot)$ and dominated $\Delta^-(\cdot)$ sets defined earlier. Let first define the numbers $q^+(x)$ and $q^-(x)$ as follows:

$$q^+(x) = \sum_{l \geq t} w^t_l \frac{|\Delta^+(x,l)|^2}{n_l^2}. \quad (4)$$

$$q^-(x) = \sum_{l \leq t} w^t_l \frac{|\Delta^-(x,l)|^2}{n_l^2}. \quad (5)$$

with $\Delta^+(x,l) = \{y : y \in Cl_l \cap \Delta^+(x)\}$ and $\Delta^-(x,l) = \{y : y \in Cl_l \cap \Delta^-(x)\}$. The set $\Delta^+(x,l)$ (resp. $\Delta^-(x,l)$) represents the set of decision objects from decision class Cl_l that are dominating (resp. dominated by) decision object x. The measures $q^+(x)$ and $q^-(x)$ can been seen as the power of decision objects that dominating and dominated by decision object x, respectively. The use of the power function permits to ensure that the value of $q^+(x)$ and $q^-(x)$ vary exponentially with the number of objects in $\Delta^+(x,l)$ and $\Delta^-(x,l)$, respectively. The first partial score is then defined as follows:

$$s_1(x) = \frac{q^+(x)}{q^+(x) + q^-(x)}, \quad \forall x \in U. \quad (6)$$

However, this version will not solve completely the problem of ties since any two decision objects x and y belonging to the same class and verifying $\Delta^+(x,l) = \Delta^-(y,l)$ and $\Delta^+(x,l) = \Delta^-(y,l)$, for all $l \in T$, will have the same score.

To further reduce/remove the problem of ties, we can exploit partial dominance relations, i.e., dominance relations with respect to single criteria. The partial dominance relation with respect to criterion $q \in C$ is defined as follows any pair of objects x and y, $x, y \in U$:

$$\delta_q(x,y) = \begin{cases} 1, & \text{if } f(x,k) \geq f(y,q) \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

The operator "$\geq$" should be replace by "$\leq$" for cost criteria and with "$=$" for nominal criteria. Then, we define two additional measures, denoted by $p^+(x)$ and $p^-(x)$, respectively, as follows:

$$p^+(x) = \sum_{y \in Cl(x) \wedge y \neq x} |\sum_{q \in C} \delta_q(y,x)|^2. \quad (8)$$

$$p^-(x) = \sum_{y \in Cl(x) \wedge y \neq x} |\sum_{q \in C} \delta_q(x,y)|^2. \quad (9)$$

As discussed previously, the use of the power function enables us to ensure that the value of $p^+(x)$ and $p^-(x)$ vary exponentially with the number of objects in $\delta(\cdot,\cdot)$. The second partial score is then defined as follows:

$$s_2(x) = \frac{p^+(x)}{p^+(x) + p^-(x)}, \quad \forall x \in U. \quad (10)$$

D. Scoring Function

The scoring function is defined as follows:

$$s : U \rightarrow \mathbb{R}^+ \quad (11)$$
$$x \rightarrow s(x) \quad (12)$$

with

$$s(x) = \pi(x) + \sum_{i=1}^{2} s_i(x). \quad (13)$$

It is important to notice that ties still may exist. This holds when: (i) two different objects from the same class have exactly the same scores on the attributes; (ii) two objects are from the same class and for which $\sum_q \delta_q, y) = \sum_q \delta_g(y,x)$; and (iii) two objects are from the same class and differ only with respect to nominal criteria. The first case seems to be consistent since the two objects should naturally have the same rank. The next two cases may be handled by considering the difference between the evaluations of objects with respect to condition criteria. This is outside the scope of this paper.

IV. Ranking Function

The ranking function is simply defined using the scoring function defined above. Let first mention that to be useful, the scoring function needs to ensure that the scores of any decision object x must be strictly higher to the score of any decision object y for all $x, y \in U$ and $Cl(x) \succ Cl(y)$. The scoring function $s(.)$ defined earlier meets this requirement, as stated by the following theorem.

Theorem 1: Let $x, y \in U$ with $Cl(x) \succ Cl(y)$. Then, the following holds:

$$s(x) > s(y). \quad (14)$$

Proof 1: By definition we have: $s(x) = \pi_{Cl(x)} + s_1(x) + s_2(x)$ and $s(y) = \pi_{Cl(y)} + s_1(y) + s_2(y)$. Accordingly, $s(x) - s(y) = \pi_{Cl(x)} - \pi_{Cl(y)} + (s_1(x) + s_2(x) - s_1(y) - s_2(y))$. Based on the definition of the absolute weights in Equation (2), we have $\pi_{Cl(x)} \succ \pi_{Cl(y)} + 2$. Furthermore and since we have $0 \leq s_1(x), s_2(x), s_1(y), s_2(y) \leq 1$, then we get $-2 \leq (s_1(x) + s_2(x) - s_1(y) - s_2(y)) \leq 2$. This ensures that $s(x) - s(y) > 0$, which means that $s(x) > s(y)$.

We can then specify the ranking function as follows:

$$r(x, y) \quad \begin{cases} >, & s(x) > s(y), \\ =, & s(x) = s(y), \\ <, & s(x) < s(y), \end{cases} \tag{15}$$

V. Application

In this section, we present the application of the proposed approach using green car data from Taiwan.

A. Dataset

The dataset used as input consists on a set of 114 green car models with respect to a collection of attributes given Table I. The description of this table is straightforward. We just mention that direction of preference of attributes maybe defined as: (i) gain if the preference is increasing with the attribute values; (ii) cost if the preference is decreasing with the attribute values; or (iii) if neither (i) and (ii) can be applied. We note also that the currency unit is TWD and prices are presented by unit and the fuel efficiency unit is Km/Ltr. The assessment of the green car models with respect to the attributes is given in Table II. The last column of this table indicates classes of considered green car models. The assignment of green car models to 10 classes have been specificated by the policy makers based on official standard used by the Taiwan's government. Excepting data tables, the classes in the rest of the paper will be denoted by Cl_1 to Cl_{10} with Cl_{10} is the most preferred class.

TABLE I
LIST OF ATTRIBUTES

Code	Name	Data Type	Unit	Preference
A1	Car brand	Symbolic		None
A2	Car model	Symbolic		None
A3	Car market price	Continuous	10K TWD	Cost
A4	Swept volume	Continuous	CC	Cost
A5	Fuel consumption	Continuous	KM/LTR	Gain
A6	Car horsepower	Continuous		Cost
A7	Car torque	Continuous		Cost
A8	Fuel Tax	Continuous	TWD	Cost
A9	Car Tax	Continuous	TWD	Cost

B. Application and Results

The application of the proposed of scoring and ranking function in the dataset in Table II is summarized in Table IV. In what follows, we illustrate the calculation of partial score $_2(x_5)$ of car #5. Based on Equation (2), the absolute weights for decision classes $Cl_1, \cdots, C_{10}$ are as follows:

Class (Cl_t)	Cl_1	Cl_2	Cl_3	Cl_4	Cl_5
Absolute weight ($\pi_{Cl_{(t)}}$)	1	5	13	24	38
Class (Cl_t)	Cl_6	Cl_7	Cl_8	Cl_9	Cl_{10}
Absolute weight ($\pi_{Cl_{(t)}}$)	56	82	113	152	224

Since cars x_5 and x_{61} belong to classes Cl_1 and Cl_8, respectively, their respective scores are defined as follows:

- $s(x_5) = \pi_{Cl_1} + s_1(x_5) + s_2(x_5)$.

The value of π_{Cl_1} corresponds to the absolute weights for decision class Cl_1. In the rest of this section, we illustrate calculation of $s_2(x_5)$. Let first calculate $p^-(x_5)$ and $p^+(x_5)$. First, we need to use Equation (7) to compute the partial dominance relations between car x_5 and cars $y \in Cl_1 \backslash \{x_5\} =$

TABLE II
DATASET

#	A1	A2	A3	A4	A5	A6	A7	A8	A9	Class
1	5	9	106.9	2494	19.6	160	21.7	7200	15210	9
2	5	11	139.9	2494	19.6	160	21.7	7200	15210	9
3	2	15	333	2143	21.6	204	51	6180	11230	9
4	2	39	551	3498	13.6	306	37.7	8640	28220	6
5	3	8	586	4395	6.3	555	69.3	11220	46170	1
6	3	21	239	647	91.9	34	5.61	4320	4320	10
7	3	22	989	1499	47.6	231	32.6	4800	7120	9
8	6	14	176	1798	23.3	99	14.5	4800	7120	9
9	6	16	202	2494	18.5	160	21.7	7200	15210	9
10	6	17	236	2494	18.8	181	22.5	7200	15210	9
11	6	19	337	3456	16	292.4	35.9	8640	28220	8
12	6	26	190	2494	19.9	181	22.5	7200	15210	9
13	6	28	679	4969	10.1	394.4	53	12180	46170	3
14	6	30	213	2494	17.1	155	21.4	7200	15210	8
15	7	31	199	1998	54.6	121	19.4	6180	11230	9
16	8	23	225	3498	13.2	302	35.7	8640	28220	6
17	8	24	321	2488	14	230	34	7200	15210	7
18	10	32	630	2995	29.7	333	44.9	7200	15210	9
19	11	33	125.9	1798	25	99	14.5	4800	7120	9
20	11	34	84.9	1497	26.5	73	11.3	4800	7120	9
21	12	41	329	2995	8.9	333	44.9	7200	15210	2
22	5	9	106.9	2494	19.6	160	21.7	7200	15210	9
23	5	10	139.9	2494	19.6	160	21.7	7200	15210	9
24	1	3	294	1984	13.8	211	35.7	6180	11230	6
25	2	39	535	3498	13.6	306	37.7	8640	28220	6
26	3	8	579	4395	6.3	555	69.3	11220	46170	1
27	3	21	299	647	91.9	34	5.61	4320	4320	10
28	6	14	159	1798	23.3	99	14.5	4800	7120	9
29	6	16	179	2494	18.5	160	21.7	7200	15210	9
30	6	17	209	2494	18.8	181	22.5	7200	15210	9
31	6	19	299	3456	16	292	35.9	8640	28220	8
32	6	26	169	2494	19.9	181	22.5	7200	15210	9
33	6	28	601	4969	10.1	394.4	53	12180	46170	3
34	6	30	189	2494	17.1	155	21.4	7200	15210	8
35	7	31	87.9	2359	14	172	24	6180	11230	7
36	8	23	225	3498	13.2	302	35.7	8640	28220	6
37	9	1	152	1997	15.15	163	34.7	3708	11230	7
38	9	2	158.8	1997	16.4	163	34.7	3708	11230	8
39	10	32	630	2995	29.7	333	44.9	7200	15210	9
40	11	34	86.9	1497	26.5	73	11.3	4800	7120	9
41	12	41	329	2995	9.9	333	44.9	7200	15210	3
42	5	9	106.9	2494	19.4	160	21.7	7200	15210	9
43	5	10	139.9	2494	19.4	160	21.7	7200	15210	9
44	1	3	291	1984	13.8	211	35.7	6180	11230	6
45	1	37	276	1984	13	211	35.7	6180	11230	6
46	2	39	576	3498	12.3	279	35.7	8640	28220	5
47	3	3	312	2979	16.9	306	40.8	7200	15210	8
48	3	8	579	4395	6.3	555	69.3	11220	46170	1
49	6	14	149	1798	22.7	99	14.5	4800	7120	9
50	6	16	179	2494	19.5	160	21.7	7200	15210	9
51	6	17	223	2500	10.7	209	25.8	7200	15210	4
52	6	19	283	3456	16.3	292.4	35.9	8640	28220	8
53	6	26	169	2494	20.5	181	22.5	7200	15210	9
54	6	28	564	5000	11.3	290	53	12180	46170	4
55	11	34	86.9	1497	25.6	73	11.3	4800	7120	9
56	12	41	326.8	2995	12.2	333	333	7200	15210	5
57	3	7	672	4395	9	44	66.3	11220	46170	2
58	3	8	579	4395	6.3	555	69.3	11220	46170	1
59	6	14	149	1798	22.7	99	14.5	4800	7120	9
60	6	16	179	2494	19.5	160	21.7	7200	15210	9
61	6	18	283	3456	16.3	292.4	35.9	8640	28220	8
62	6	20	180	2362	14.3	150	19.1	8720	7120	7
63	11	34	85.9	1497	25.6	73	11.3	4800	7120	9
64	12	41	326.8	2995	12.2	333	44.9	7200	15210	5
65	2	39	576	3498	12.3	279	35.7	8640	28220	5
66	3	6	672	4395	9	449	66.3	11220	46170	2
67	3	8	612	4395	8.6	408	61.2	11220	46170	2
68	4	12	139	1500	21	113	14.7	4800	7120	9
69	4	25	99.9	1300	20.8	88	12.3	4800	7120	9
70	6	14	139	1800	22.7	99	14.5	4800	7120	9
71	6	18	287	3500	13.8	307	37.9	8640	28220	6
72	6	20	180	2362	14.3	150	19.1	8720	7120	7
73	6	27	564	5000	11.3	290	53	12180	46170	4
74	11	36	131	1798	26.3	99	14.5	4800	7120	9

$\{x_{26}, x_{48}, x_{58}\}$. The result of this step is summarized in Table III. Then, based on Equations (8) and (9), we obtain:

$$p^+(x_5) = \sum_{y \in \ Cl(x_5) \wedge y \neq x_5} |\sum_{q \in C} \delta_q(y, x_5)|^2 = 3 \cdot 8^2 = 192.$$

$$p^-(x_5) = \sum_{y \in \ Cl(x_5) \wedge y \neq x_5} |\sum_{q \in C} \delta_q(x_5, y)|^2 = 3 \cdot 7^2 = 147.$$

Then, by using Equation (16) we get:

$$s_2(x_5) = \frac{192}{147 + 192} \tag{16}$$

$$= 0.56637. \tag{17}$$

TABLE III
PARTIAL DOMINANCE RELATION FOR CAR x_5 WITH RESPECT TO CARS x_{26}, x_{48} AND x_{58}

x	y	$\delta_q(x, y)$							
		1	2	3	4	5	6	7	8
x_5	x_{26}	1	1	0	1	1	1	1	1
x_5	x_{48}	1	1	0	1	1	1	1	1
x_5	x_{58}	1	1	0	1	1	1	1	1
x_{26}	x_5	1	1	1	1	1	1	1	1
x_{48}	x_5	1	1	1	1	1	1	1	1
x_{58}	x_5	1	1	1	1	1	1	1	1

VI. CONCLUSION

The paper proposes a score-based ranking algorithm permitting to transform the output of ordinal classification into a ranking while respecting the preference order established by the classes. The scores are defined using elementary preference information extracted from the dataset. This means that no additional information are requested from the decision maker. This is an important feature of the proposed scoring function, particulary for large datasets. The proposed approach is illustrated using green car data in Taiwan.

As mentioned in Section III-D, ties still can exist in some specific cases. To address this issue, we intent to extend the scoring function by adding a third partial score relying on the difference between the evaluations of objects with respect to condition criteria. Future work will also consider the application of the proposed approach using other real-life datasets. Additionally, we intend to investigate the performance and computational behavior of proposed algorithms with large to very-large datasets.

REFERENCES

[1] Shereen Fouad and Peter Tino. Adaptive Metric Learning Vector Quantization for Ordinal Classification. *Neural Computation*, 24(11):2825–2851, 11 2012.

[2] Eibe Frank and Mark Hall. A simple approach to ordinal classification. In Luc De Raedt and Peter Flach, editors, *Machine Learning: ECML 2001*, pages 145–156, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[3] B. Ghaddar, N. Sakr, and Y. Asiedu. Spare parts stocking analysis using genetic programming. *European Journal of Operational Research*, 252(1):136–144, 2016.

[4] S. Greco, B. Matarazzo, and R. Slowiński. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, 129(1):1–47, 2001.

[5] P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez. Ordinal regression methods: Survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146, 2016.

[6] Q. Hu, S. Chakhar, S. Siraj, and A. Labib. Spare parts classification in industrial manufacturing using the dominance-based rough set approach. *European Journal of Operational Research*, 262(3):1136–1163, 2017.

[7] L. Polkowski and P. Artiemjew. *Granular Computing in Decision Approximation: An Application of Rough Mereology*, volume 77 of *Intelligent Systems Reference Library*, chapter Learning Data Classification: Classifiers in General and in Decision Systems, pages 33–62. Springer International Publishing, Cham, Switzerland, 2015.

[8] R. Slowiński, S. Greco, and B. Matarazzo. Rough set analysis of preference-ordered data. In J.J. Alpigini, J.F. Peters, A. Skowron, and N. Zhong, editors, *Rough Sets and Current Trends in Computing*, volume 2475 of *Lecture Notes in Artificial Intelligence*, pages 44–59. Springer Berlin Heidelberg, 2002.

[9] R. Słowiński, S. Greco, and B. Matarazzo. Rough sets in decision making. In R.A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 7753–7787. Springer, New York, 2009.

[10] X. Zhang and J. Zeng. Joint optimization of condition-based opportunistic maintenance and spare parts provisioning policy in multiunit systems. *European Journal of Operational Research*, 262(2):479–498, 2017.

TABLE IV
SCORES AND RANKS OF GREEN CARS

x	$\pi_{Cl(x)}$	$q^{-}(x)$	$q^{+}(x)$	$s_1(x)$	$p^{-}(x)$	$p^{+}(x)$	$s_2(x)$	$s(x)$	Rank
1	152	0.52520	0.00192	0.00365	698	812	0.53775	152.54139	19
2	152	4.32497	0.00021	0.00005	625	865	0.58054	152.58059	16
3	152	7.48783	0.00021	0.00003	193	657	0.77294	152.77297	5
4	56	9.60074	0.02512	0.00261	92	253	0.73333	56.73594	50
5	1	25.12530	0.19381	0.00765	147	192	0.56637	1.57403	71
6	224	26.09350	0.04845	0.00185	64	49	0.43363	224.43548	2
7	152	36.35710	0.00021	0.00001	302	530	0.63702	152.63703	12
8	152	44.74190	0.00342	0.00008	1016	539	0.34662	152.34670	23
9	152	53.99710	0.00342	0.00006	598	1110	0.64988	152.64995	11
10	152	64.12260	0.00085	0.00001	370	1218	0.76700	152.76702	6
11	113	66.91020	0.03925	0.00059	127	327	0.72026	113.72085	37
12	152	86.98450	0.00192	0.00002	419	1183	0.73845	152.73847	8
13	13	93.75230	0.12404	0.00132	49	100	0.67114	13.67246	64
14	113	100.76700	0.01744	0.00017	285	88	0.23592	113.23610	41
15	152	127.80500	0.00021	0	525	493	0.48428	152.48428	22
16	56	115.80800	0.02512	0.00022	184	161	0.46667	56.46688	52
17	82	143.13100	0.01340	0.00009	3	111	0.97368	82.97378	44
18	152	197.04300	0.00085	0	126	1158	0.90187	152.90187	3
19	152	232.94500	0.00021	0	970	405	0.29455	152.29455	30
20	152	262.79300	0.00021	0	1251	202	0.13902	152.13902	34
21	5	362.26600	0.09515	0.00026	97	1	0.01020	5.01047	70
22	152	376.65200	0.00192	0.00001	698	812	0.53775	152.53775	20
23	152	453.31100	0.00085	0	640	880	0.57895	152.57895	17
24	56	387.01700	0.02512	0.00006	232	122	0.34463	56.34470	54
25	56	419.29500	0.00628	0.00001	107	238	0.68986	56.68987	51
26	1	861.95100	0.10902	0.00013	192	177	0.47967	1.47980	72
27	224	845.99800	0.19381	0.00023	49	64	0.56637	224.56660	1
28	152	777.66700	0.00192	0	1057	518	0.32889	152.32889	24
29	152	863.95300	0.00192	0	657	1075	0.62067	152.62067	13
30	152	1037.42000	0.00021	0	385	1203	0.75756	152.75756	7
31	113	1014.35000	0.00436	0	160	264	0.62264	113.62265	40
32	152	1241.46000	0.00085	0	479	1116	0.69969	152.69969	9
33	13	1238.22000	0.03101	0.00003	64	85	0.57047	13.57049	65
34	113	1309.45000	0.00436	0	300	73	0.19571	113.19571	42
35	82	1262.72000	0.01340	0.00001	63	34	0.35052	82.35053	47
36	56	1249.11000	0.02512	0.00002	184	161	0.46667	56.46669	53
37	82	1391.49000	0.01340	0.00001	59	28	0.32184	82.32185	48
38	113	1582.60000	0.00436	0	212	8	0.03636	113.03637	43
39	152	1853.98000	0.00085	0	126	1158	0.90187	152.90187	4
40	152	1969.33000	0.00342	0	1221	247	0.16826	152.16826	31
41	13	1860.07000	0.06718	0.00004	72	0	0	13.00004	66
42	152	2219.05000	0.00192	0	698	812	0.53775	152.53775	21
43	152	2400.47000	0.00085	0	640	880	0.57895	152.57895	18
44	56	1928.99000	0.00628	0	247	107	0.30226	56.30226	55
45	56	2010.17000	0.00628	0	256	80	0.23810	56.23810	56
46	38	2055.03000	0.10047	0.00005	72	96	0.57143	38.57148	57
47	113	2531.77000	0.00436	0	51	139	0.73158	113.73158	36
48	1	3792.38000	0.10902	0.00003	192	177	0.47967	1.47970	73
49	152	3259.56000	0.00085	0	1087	503	0.31635	152.31635	25
50	152	3567.78000	0.00192	0	657	1075	0.62067	152.62067	14
51	24	2989.16000	0.03774	0.00001	98	2	0.02000	24.02001	63
52	113	3494.47000	0.00436	0	173	290	0.62635	113.62635	39
53	152	4128.95000	0.00085	0	479	1116	0.69969	152.69969	10
54	24	3379.66000	0.03721	0.00001	50	98	0.66216	24.66217	61
55	152	4462.92000	0.00342	0	1221	247	0.16826	152.16826	32
56	38	3432.50000	0.10047	0.00003	81	72	0.47059	38.47062	59
57	5	4756.92000	0.01454	0	75	97	0.56395	5.56396	68
58	1	6092.11000	0.10902	0.00002	192	177	0.47967	1.47969	74
59	152	5222.01000	0.00085	0	1087	503	0.31635	152.31635	26
60	152	5610.30000	0.00192	0	657	1075	0.62067	152.62067	15
61	113	5298.05000	0.00436	0	145	264	0.64548	113.64548	38
62	82	4967.44000	0.05358	0.00001	107	83	0.43684	82.43685	45
63	152	6414.13000	0.00085	0	1236	217	0.14935	152.14935	33
64	38	5153.54000	0.02512	0	96	57	0.37255	38.37255	60
65	38	5336.42000	0.10047	0.00002	72	96	0.57143	38.57145	58
66	5	7405.33000	0.01454	0	52	134	0.72043	5.72043	67
67	5	7518.85000	0.01454	0	85	77	0.47531	5.47531	69
68	152	7597.77000	0.00021	0	799	344	0.30096	152.30096	29
69	152	7713.20000	0.00021	0	1057	141	0.11770	152.11770	35
70	152	7829.51000	0.00021	0	1066	485	0.31270	152.31270	27
71	56	5995.79000	0.01256	0	28	208	0.88136	56.88136	49
72	82	6479.99000	0.05358	0.00001	107	83	0.43684	82.43685	46
73	24	6409.38000	0.03721	0.00001	50	98	0.66216	24.66217	62
74	152	8424.09000	0.00021	0	957	418	0.30400	152.30400	28