Discovering diverse top-k characteristic lists

Antonio Lopez-Martinez-Carrasco[1(✉)], Hugo M. Proença[2], Jose M. Juarez[1], Matthijs van Leeuwen[2], and Manuel Campos[1,3]

[1] AIKE Research Group (INTICO), University of Murcia, Spain
[2] Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
[3] Murcian Bio-Health Institute (IMIB-Arrixaca), Spain

Abstract. In this work, we define the new problem of finding diverse top-k characteristic lists to provide different statistically robust explanations of the same dataset. This type of problem is often encountered in complex domains, such as medicine, in which a single model cannot consistently explain the already established ground truth, needing a diversity of models. We propose a solution for this new problem based on Subgroup Discovery (SD). Moreover, the diversity is described in terms of coverage and descriptions. The characteristic lists are obtained using an extension of SD, in which a subgroup identifies a set of relations between attributes (description) with respect to an attribute of interest (target). In particular, the generation of these characteristic lists is driven by the Minimum Description Length (MDL) principle, which is based on the idea that the best explanation of the data is the one that achieves the greatest compression. Finally, we also propose an algorithm called GMSL which is simple and easy to interpret and obtains a collection of diverse top-k characteristic lists.

Keywords: Subgroup Discovery · Subgroup List · the Minimum Description Length principle · Algorithm · Interpretable Machine Learning.

1 Introduction

More and more often, Artificial Intelligence is required to generate readable, understandable and transparent models. In contrast to black-box machine learning (e.g., neural network models), interpretable machine learning is an increasing trend whose objective is to develop new methods and tools which allow humans to understand machine learning models and to interpret their results in many critical areas, such as medicine or economy. In this context, different research has been carried out [12, 13].

The discovery of a collection of descriptions is helpful to better understand datasets. In this context, one type of collection is the characteristic list, whose purpose is to generalise an individual belonging to a specific category. These descriptions explain all typical features that characterize the individuals belonging to a specific category for a descriptive purpose [1]. An example of a description from a characteristic list is shown in Figure 1.

Fig. 1. Example of a description from a characteristic list.

The utilization of multiple characteristic lists is relevant because a single explanation of a target value is not always enough. A clear example of both needs is the clinical domain, in which a patient could have some diagnosis. A relevant task is to find all the risk factors that differentiate a diagnosis from others, not only from a predictive point of view, but from the descriptive point of view. However, a single characteristic list provides a limited explanation, having little value due to the possible lack of meaning from the clinical point of view. For example, a characteristic list automatically generated by some machine learning algorithm could not make sense for the clinicians and, therefore, be discarded.

Subgroup Discovery (SD) can be used as building block of characteristic lists. A subgroup identifies a relation between attributes (description) and an attribute of interest (target). Besides, subgroups can be used as local descriptive models that characterize subpopulations, in contrast to the whole population, in relation to the target attribute given a quality measure. However, only sets with few subgroups can be easily interpreted by an expert. To solve this problem, we can build a subgroup list model. We illustrate the advantages in Figure 2, showing subgroups and subgroup lists extracted from the *zoo* dataset. On the one hand, a subgroup contains a set of selectors (i.e., a pattern or description) and it is generated when a quality measure given, e.g. Weighted Relative Accuracy (WRAcc), with respect to a target value is above a threshold. In the figure, the subgroup s1 "milk = yes" contains a single selector to define the class *type = 'mammal'*. On the other hand, a subgroup list is an ordered collection of subgroups that explain the target value. In the figure, an example of four subgroup lists is depicted. Note that either a subgroup or a subgroup list provides an explanation of how to define a single class (*type = 'mammal'* in this example), but not the others. Moreover, it is readable, understandable and has the potential to be interpretable.

A subgroup list can be interpreted as a decision list, since it is an ordered collection of subgroups of the form "else-if" (i.e., a subgroup description is only reached if all the above ones are not being true). Another model that can be used for the proposed problem is the decision set, which is formed by an unordered collection of subgroups of the form "if" (i.e., all subgroup descriptions apply independently). Although our objective is to describe data, both models can be used either for description or prediction tasks [7].

A great difficulty when creating a subgroup list is the large number of subgroups that are extracted. In the example shown in Figure 2, there are 8,537,383 subgroups mined with an exhaustive SD algorithm that could be used to create subgroup lists. This can be solved using the Minimum Description Length

(MDL) principle [6], a method of inductive inference whose fundamental idea is that the best explanation of the data is the one that achieves the greatest compression. In the context of SD and subgroup lists, it was shown that using the MDL principle is equivalent to performing a Bayesian statistical test and multiple hypothesis testing correction for every subgroup [11, 10]. Thus, this leads to the discovery of statistically robust subgroups and subgroup lists.

	Subgroup description	type= 'mammal' (Pos)	type= 'other' (Neg)
s1	milk='yes'	41	0
s2	venomous='no'	0	52
dr	-	0	8

	Subgroup description	type= 'mammal' (Pos)	type= 'other' (Neg)
s1	backbone='yes', hair = 'yes'	39	0
s2	fins='no'	0	47
dr	-	2	13

	Subgroup description	type= 'mammal' (Pos)	type= 'other' (Neg)
s1	eggs='no'	40	2
s2	backbone='yes'	1	41
s3	feathers='no'	0	17
dr	-	0	0

	Subgroup description	type= 'mammal' (Pos)	type= 'other' (Neg)
s1	hair='yes'	39	4
s2	breathes='yes'	2	35
s3	toothed='yes'	0	14
dr	-	0	7

Fig. 2. An example of four subgroup lists generated from the *zoo* dataset (i.e., four different explanations of this dataset), being the target *type = 'mammal'*.
Notation: s1: subgroup1; dr: default rule; pos: positives; neg: negatives.

The main contributions of this research are: (1) the definition of the new problem of finding diverse top-k characteristic lists, and (2) a new algorithm called GMSL that solves this problem by using SD, the subgroup list model, and the MDL principle. Moreover, its results are simple, readable, understandable and statistically robust at the same time. This contribution improves the state of the art, since existing algorithms generate only one subgroup list and, therefore, different explanations for the same data are not possible. Note that, to the best of our knowledge, no algorithm in the literature combines SD and the MDL principle to generate diverse top-k subgroup lists.

The remainder of this paper is structured as follows: Section 2 defines the problem tackled in this research, while Section 3 shows and explains our proposal: the new algorithm called GMSL that generates diverse top-k subgroup lists. Moreover, Section 4 describes the configuration of the experiments carried out in this work and provides a discussion of the results obtained. Finally, Section 5 presents the conclusions reached after carrying out the research.

2 Problem statement and background

This section formalizes the problem tackled in this research, i.e., the generation of diverse top-k characteristic lists by using Subgroup Discovery (SD), the subgroup list model, and the Minimum Description Length (MDL) principle.

2.1 The problem of discovering diverse top-k characteristic lists

The fundamental concepts of this new problem are defined in this section.

First, an attribute a is a relation between an object property and its value. For example, $a = hair : no$. Moreover, the set of all unique values that an attribute can take is defined as the domain of the attribute and is denoted as $dom(a)$. Note that, depending on its domain, an attribute can be nominal or numeric. Second, an instance i is a tuple $i = (a_1, \ldots, a_m)$ of attributes, for example, $i = (milk : yes, hair : yes)$. Finally, a dataset d is a tuple $d = (i_1, \ldots, i_n)$ of instances. For example, $d = ((milk : yes, hair : yes), (milk : yes, hair : no))$. Note that we use the notation $v_{x,y}$ to indicate the value of the x-th instance i_x and of the y-th attribute a_y from a dataset d.

According to these basic definitions, the following ones can be given:

Definition 1 (Selector e). *Given an attribute a_y from a dataset d, a binary operator $\in \{=, \neq, <, >, \leq, \geq\}$ and a value w, being w in the domain of a_y, then a selector e is defined as a 3-tuple of the form $(a_y, operator, w)$.*

Informally, this means that a selector is a binary relation between an attribute from a dataset and one of its possible values, representing a property of a subset of instances from this dataset. Some examples of selectors are $e_1 = (age, >, 50)$ and $e_2 = (venomous, =, yes)$.

Definition 2 (Selector coverage). *Given an instance i_x, an attribute a_y and a selector $e = (a_y, operator, w \in dom(a_y))$, then i_x is covered by e if the binary expression "$v_{x,y}$ operator w" holds true. Otherwise, we say that it is not covered by e.*

Definition 3 (Pattern p). *A pattern p is a list of selectors $< e_1, \ldots, e_x >$ (i.e., a conjunction) in which all attributes of the selectors are different.*

Informally, this means that a pattern represents a list of properties of a subset of instances from a dataset.

Definition 4 (Pattern coverage). *Given an instance i and a pattern p, then i is covered by p if i is covered by e_x, $\forall e_x \in p$. Otherwise, we say that it is not covered by p.*

Definition 5 (Characteristic list l). *Given a dataset d and a selector e (denominated as category), then a characteristic list l is a collection of patterns $< p_1, \ldots, p_y >$ (each of them is denominated as "description") that allow to describe the instances from d belonging to e. Note that a characteristic list is used for a descriptive purpose.*

An example of description from a characteristic list is depicted in Figure 1. This description is formed by a set of selectors that allow to describe the individuals belonging to the category *animal = 'turtle'*.

It is necessary to explain why "top-k" and "diversity" properties are essential in the new problem defined. Firstly, it is focused only on the generation of the top-k characteristic lists due to this generation is limited by the available computational capacity. This means that it is not feasible to carry out an exhaustive generation of all possible characteristic lists. Secondly, diversity is essential in this case, since multiple characteristic lists from l_1 to l_k will be generated and, therefore, it is necessary to ensure that they will be different and non-redundant. Diversity can be achieved both in terms of coverage and descriptions. The diversity in terms of coverage is considered when building a single characteristic list l_x to minimize the number of instances already covered by previous patterns. This means that, given two patterns $p_a \in l_x$ and $p_b \in l_x$, the instances covered by both patterns at the same time should be as few as possible. The diversity in terms of descriptions implies using different selectors and patterns in the different characteristic lists to ensure that the models provide multiple explanations of the same category or target value. This means that, given two characteristic lists l_x and l_y, then $\forall p_a$, *if* $p_a \in l_x$, *then* $p_a \notin l_y$.

Therefore, the new problem of discovering diverse top-k characteristic lists is defined as follows:

Definition 6 (Discovering diverse top-k characteristic lists problem). *Given a dataset d, a category e (in form of a selector) and the k maximum number of characteristic lists to generate, then the problem of discovering diverse top-k characteristic lists consists of generating a collection of characteristic lists $< l_1, \ldots, l_k >$ such that they are diverse and represent different explanations or perspectives of d in relation to e.*

Finally, the proposal carried out in this work (i.e., GMSL algorithm, which is explained in Section 3) solves this problem by using SD, the subgroup list model, and the MDL principle.

2.2 Subgroup Discovery

SD [2] is a supervised machine learning technique whose purpose is the identification of a set of relations between attributes (denominated as *description*) with respect to an attribute of interest (denominated as *target*). This technique is widely used for exploratory and descriptive data analysis and is also useful for obtaining general relations in a dataset and automatically generating hypotheses. In particular, SD helps to obtain groups of individuals that might overlap. However, as with many pattern mining techniques, SD experiences some problems such as pattern explosion or lack of statistical guarantees specially when using datasets with many attributes [9]. Therefore, configuring a list of the best subgroups that faithfully describes a dataset is not trivial.

Additionally, the fundamental concepts of SD are described as follows:

Given a pattern p and a selector e, a subgroup s is a pair (p, e) in which the pattern is denominated as 'description' and the selector is denominated as 'target'. Subgroups can be used for either a predictive purpose or a descriptive purpose (i.e., characteristic subgroups) [1]. Therefore, since our objective in this research is to describe and characterize individuals from a dataset, subgroups will be used as a fundamental part of characteristic lists (subgroup lists, in this case) with the objective of identifying all properties related to a specific category or target attribute. An example of subgroup is $s = (< (shell, =, yes), (feathers, =, no), (backbone, =, yes) >, (turtle, =, yes))$. Finally, given a subgroup s and a dataset d, a quality measure q is a function that computes one numeric value according to s and to certain characteristics from d [4]. Some examples of quality measures are Sensitivity, Piatetsky Shapiro or Weighted Relative Accuracy (WRAcc).

Following these definitions, given a dataset d, a quality measure q and a numeric value *threshold*, the subgroup discovery problem consists of exploring the search space of d in order to generate subgroups that have a value of q above *threshold*. Formally: $\mathcal{R} = \{(s, quality\_value) | quality\_value \geq threshold\}$.

Some examples of algorithms that generate individual subgroups are SD-Map [3], CN2-SD [8] or ID-Rsd [5], among others.

2.3 The Subgroup List model

The subgroup list model was initially presented in [11] and, afterwards, expanded and detailed in [10]. A subgroup list is a collection of ordered subgroups followed by a default rule, whose objective is to partition the input data and to provide a description (i.e., an individual subgroup) of each of these partitions, except the last one (that corresponds to the default rule). While the default rule represents the dataset average and covers the instances that are well described by the dataset distribution, the subgroups cover the instances that are statistically different and interesting, compared to dataset distribution. Therefore, each instance of the input dataset can only be covered either by one individual subgroup or by the default rule. For example, if a subgroup list contains 10 subgroups, this means that the input dataset was partitioned into 11 subsets: the first 10 of them correspond to the 10 individual subgroups and the last one corresponds to the default rule. An example of subgroup list is shown in Figure 3.

$subgroup_1$:	IF	$description_1$	THEN	$distribution_1(target)$
$subgroup_2$:	ELSE IF	$description_2$	THEN	$distribution_2(target)$
		$\vdots$		
$subgroup_w$:	ELSE IF	$description_w$	THEN	$distribution_w(target)$
dataset:	ELSE			$distribution(target)$

Fig. 3. Example of subgroup list with w subgroups.

2.4 The MDL principle for discovering a single subgroup list

According to the MDL principle, the best individual subgroup list is the one that compresses the data and the model the most, i.e., the simplest subgroup list that best fits the data. The authors of [10] defined the MDL encoding of the optimal subgroup list for a certain dataset.

However, as the problem of finding an optimal subgroup list is NP-hard, the authors of [10] also proposed a greedy approach that iteratively added one subgroup at the time to the subgroup list (after the last subgroup and before the default rule). According to this, given a dataset d, a subgroup list model M, and a subgroup candidate s, the best subgroup to add to a single subgroup list is the one that maximize the compression gain, which is defined as follows:

$$\Delta_\beta L(d, M \oplus s) = \frac{L(d, M) - L(d, M \oplus s)}{(n_s)^\beta} + \frac{L(M) - L(M \oplus s)}{(n_s)^\beta} \quad (1)$$

Note that the $\oplus$ operator represents adding s at the end of M (before the default rule), and n_s is the number of instances covered by the description of s.

More details about $\Delta_\beta L$ and the β parameter can be found in [10], although the intuition is as follows: (1) a subgroup candidate that maximizes $\Delta_\beta L$ is maximizing a Bayesian proportions tests between the subgroup distribution and the dataset distribution while penalizing for larger descriptions; (2) $\Delta_\beta L > 0$ means there is more statistical evidence in favour of adding the subgroup candidate to the list than not adding it; and (3) β values closer to 0 prioritize subgroup candidates that cover more instances, while β values closer to 1 prioritize subgroup candidates that cover less instances.

Currently, state of the art only focuses on algorithms to discover a single subgroup list (e.g., SDD++ algorithm [10]). Therefore, they cannot return diverse top-k subgroup lists automatically.

3 GMSL algorithm

In this work, we propose the Generation of Multiple Subgroup Lists algorithm (GMSL), whose purpose is to generate diverse top-k Subgroup Lists by combining SD and the MDL principle.

Our proposal is detailed in Algorithm 1, and it requires the following inputs: a dataset d, a collection of subgroup candidates $\mathcal{C}$, the maximum number of subgroup lists to generate, and the normalization parameter β used by the compression gain $\Delta_\beta L$ (see Equation 1). Besides, it is also necessary to state that the subgroup candidates from $\mathcal{C}$ could be generated with any algorithm and could be also filtered before executing GMSL algorithm.

The algorithm starts with the creation of the list $\mathcal{L}$, which has size max_sl and is initialized with empty subgroup lists (line 1). Next, we iterate through $\mathcal{L}$ (loop of the line 2), and for each subgroup list, continuous iterations through $\mathcal{C}$ are carried out in order to find the best subgroup candidate to add (lines 6 - 12). The compression gain for each current subgroup candidate is calculated with the

Algorithm 1 GMSL algorithm.

Input: d { dataset } ; $\mathcal{C}$ { subgroup candidates } ; max_sl { maximum number of subgroup lists to generate ($\mathbb{N}$) } ; β { normalization parameter $\in [0, 1]$ }
Output: $\mathcal{L}$: collection of subgroup lists.

```
1: L := create a collection with max_sl empty subgroup lists.
2: for each sl ∈ L do
3:     repeat
4:         best_candidate := NULL
5:         bc_comp_gain := 0
6:         for each current_candidate ∈ C do
7:             cc_comp_gain := Δ_β L(d, sl ⊕ current_candidate)
8:             if cc_comp_gain > bc_comp_gain then
9:                 best_candidate := current_candidate
10:                bc_comp_gain := cc_comp_gain
11:            end if
12:        end for
13:        if best_candidate ≠ NULL then
14:            sl := sl ⊕ best_candidate
15:            C.delete(best_candidate)
16:            C.deleteRefinements(best_candidate)
17:        end if
18:    until best_candidate = NULL
19: end for
20: return L
```

compression gain $\Delta_\beta L$ (line 7). The candidate with the highest compression gain will be selected (lines 8 - 11) and added to the current subgroup list (lines 13 - 17) until there are no subgroup candidates with positive compression gain. Finally, the algorithm returns the collection $\mathcal{L}$ containing max_sl subgroup lists. Note that computing the compression gain for each subgroup candidate using the MDL principle guarantees that all subgroups added to a subgroups list are statistically robust. Moreover, it is also relevant to remark that GMSL algorithm also encourages the generation of diverse subgroup lists to allow different explanations of the dataset.

Finally, we have to highlight how the algorithm generates diverse subgroup lists. In the first place, diversity in terms of coverage is guaranteed due to the utilization of the subgroup list model, since each instance of the input dataset can only be covered either by one individual subgroup or by the default rule. In the second place, diversity in terms of descriptions is achieved because each time that a subgroup candidate from $\mathcal{C}$ is added to a subgroup list, that subgroup and its refinements are deleted (lines 15 and 16). Therefore, each subgroup candidate appears at most once and the appearance of the same selectors in the different patterns is also minimized.

4 Experiments and Discussion

GMSL algorithm was implemented in `subgroups` python library[4]. The goal of the experiments carried out in this work was to validate our proposal in relation to the new defined problem (i.e., to verify whether GMSL algorithm can generate diverse top-k characteristic lists in form of subgroup lists). We used for this purpose the well-known *car-evaluation* dataset from UCI repository with *class = 'acc'* as target, meaning that the car is *acceptable* to be bought. The One Hot Encoding technique was applied to the dataset with the objective that attributes were binary. Therefore, this dataset had 1,728 instances and 18 attributes. After that, an exhaustive SD algorithm was executed using WRAcc quality measure and a threshold value of 0 (i.e., only subgroups whose WRAcc quality measure value is greater or equal than 0 were generated) and a maximum depth of 2. Note that any exhaustive SD algorithm could be applied in this point, since subgroups obtained by any exhaustive SD algorithm are always the same as long as the same quality measure and parameters are used. Finally, 302 subgroups were obtained. These subgroup candidates ($\mathcal{C}$) were the main input of GMSL algorithm to generate diverse top-k subgroup lists.

After carrying out the experiments described, diverse top-3 subgroup lists were generated, and they are represented in Figure 4. For each one, the following elements are shown: (1) their individual subgroups and the default rule (denoted as `dr`), (2) the number of positive (i.e., such that the class is equal to 'acc') and negative (i.e., such that the class is not equal to 'acc') instances of the dataset, and (3) the cumulative sum of positive and negative instances covered by the subgroup list.

These three diverse subgroup lists shown in Figure 4 represent different explanations of the same dataset. Different subgroups (i.e., different subgroup descriptions, which use different patterns) were used in the different subgroup lists. Therefore, different and diverse explanations were generated from the same data.

The figure also shows, for example, that the first and second subgroup lists include the original attribute *buying* (*buying_vhigh* and *buying_low* after applying One Hot Encoding), which is not used by the third subgroup list. Moreover, different attributes generate from the original *doors* attribute are used by all subgroup lists. Additionally, the first and second subgroup lists have 2 subgroups whose description has a single selector, while the third subgroup list has 3 subgroups whose description has a single selector. Besides, note that subgroups in a subgroup list need to be interpreted sequentially, since a subgroup list is ordered by definition.

According to the "cusum" value of the last subgroup (i.e., before the default rule), it can be observed that the first and third subgroup lists cover more positive examples than the second subgroup list. In the same way, the first subgroup list has fewer subgroups, being more general, while the second subgroup list has more subgroups, being more specific. It is relevant to note that, while subgroups are local model, subgroup lists are global model, since they cover the whole

[4] Source code available on: https://github.com/antoniolopezmc/subgroups

	Subgroup description	Pos-Neg instances	Cusum
s1	doors_2='no', lug_boot_low='no'	384-384	384-384
s2	buying_vhigh='no'	0-720	384-1104
s3	buying_low='no'	0-240	384-1344
dr	-	0-0	384-1344

	Subgroup description	Pos-Neg instances	Cusum
s1	doors_2='no', lug_boot_high='yes'	204-180	204-180
s2	doors_2='no', lug_boot_med='no'	0-384	204-564
s3	doors_2='no', persons_small='no'	145-111	349-675
s4	persons_small='no'	0-384	349-1059
s5	lug_boot_high='yes'	0-64	349-1123
s6	buying_vhigh='yes', lug_boot_low='no'	0-48	349-1171
dr	-	35-173	384-1344

	Subgroup description	Pos-Neg instances	Cusum
s1	doors_4='yes', lug_boot_low='no'	198-186	198-186
s2	doors_more='no', lug_boot_low='no'	0-384	198-570
s3	lug_boot_low='no'	186-198	384-768
s4	maint_2='no'	0-432	384-1200
s5	doors_2='no'	0-96	384-1296
dr	-	0-48	384-1344

Fig. 4. Diverse top-3 subgroup lists generated from *car-evaluation* dataset (i.e., three different explanations of this dataset) with *class = 'acc'* as target, meaning that the car is *acceptable* to be bought.
Notation: s1: subgroup1; dr: default rule; pos: positive instances; neg: negative instances; cusum: cumulative sum of pos/neg instances.

dataset. Moreover, subgroup list model is focused on a value of a target attribute. Additionally, each subgroup list has a different number of subgroups: the first subgroup list has three, the second subgroup list has six, and the third subgroup list has five.

It is necessary to remember that the collection of subgroup candidates is generated a-priori and, then, taken as an input by GMSL algorithm. Although this could penalize the algorithm in terms of memory consumption, it is also an advantage in term of flexibility, since it allows to prefilter this collection and to introduce domain knowledge. For example, some negative subgroups such as *doors_2 = 'no'* or *doors_more = 'no'* were generated from the *car-evaluation* dataset. However, they may not make sense for the user from the logical point of view, and consequently, they could be deleted before executing GMSL algorithm.

Note that subgroups from a subgroup list do not overlap by definition [10]. However, if we analyse each of these subgroups individually (i.e., without considering the subgroup list model), they could cover the same instances of the database.

In summary, we show for a particular case study that our proposal is suitable for solving the new problem defined initially, since it can discover diverse top-k characteristic lists in form of subgroup lists using SD and the MDL principle.

5 Conclusions

In this research, we defined the novel problem of discovering diverse top-k characteristic lists, which consists of providing users with the k best and diverse explanations of a dataset with a binary-target attribute.

To solve this problem, we proposed GMSL, an algorithm that takes a set of pre-computed subgroup candidates as input and returns a collection of diverse top-k subgroup lists. The goodness of fit is measured using the MDL principle and, moreover, diversity is defined in terms of coverage and descriptions. This way, we can provide different perspectives of the same data through the diverse top-k subgroup lists.

As shown in the examples, the results are simple and can be easily interpreted. To the best of our knowledge, this is the first proposal that uses SD and the MDL principle to solve the new defined problem.

Finally, future research could extend and improve the proposed algorithm in different ways, for example, by generating subgroup lists without a collection of subgroup candidates loaded a-priori. Moreover, the overlap between subgroups from a subgroup list could be also study in order to improve the model interpretability. Additionally, it would be interesting to extend the problem to a multiclass setting.

Acknowledgments

This work was partially funded by the CONFAINCE project (Ref: PID2021-122194OB-I00) by MCIN/AEI/10.13039/501100011033 and, as appropriate, by "ERDF A way of making Europe", by the "European Union" or by the "European Union NextGenerationEU/PRTR", and by the GRALENIA project (Ref: 2021/C005/00150055) supported by the Spanish Ministry of Economic Affairs and Digital Transformation, the Spanish Secretariat of State for Digitization and Artificial Intelligence, Red.es and by the NextGenerationEU funding. Moreover, this research was also partially funded by a national grant (Ref: FPU18/02220), financed by the Spanish Ministry of Science, Innovation and Universities (MCIU) and by a mobility grant (Ref: R-933/2021), financed by the University of Murcia.

References

1. Alkhatib, A., Boström, H., Vazirgiannis, M.: Explaining predictions by characteristic rules. In: Proc. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer International Publishing (2022)
2. Atzmueller, M.: Subgroup Discovery - Advanced Review. WIREs: Data Mining and Knowledge Discovery **5**(1), 35–49 (2015)
3. Atzmueller, M., Puppe, F.: SD-Map - A fast algorithm for exhaustive subgroup discovery. In: Knowledge Discovery in Databases (PKDD 2006). pp. 6–17. Springer Berlin Heidelberg (2006)

4. Duivesteijn, W., Knobbe, A.: Exploiting false discoveries - statistical validation of patterns and quality measures in subgroup discovery. In: IEEE 11th International Conference on Data Mining (ICDM'11). pp. 151–160 (2011)
5. Grosskreutz, H., Paurat, D.: Fast and memory-efficient discovery of the top-k relevant subgroups in a reduced candidate space. In: Machine Learning and Knowledge Discovery in Databases. pp. 533–548. Springer Berlin Heidelberg (2011)
6. Grünwald, P.D.: The Minimum Description Length Principle, MIT Press Books, vol. 1. The MIT Press (December 2007)
7. Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable decision sets: A joint framework for description and prediction. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1675–1684. KDD '16, Association for Computing Machinery (2016)
8. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with CN2-SD. J. Mach. Learn. Res. **5**, 153–188 (dec 2004)
9. van Leeuwen, M., Ukkonen, A.: Expect the unexpected – on the significance of subgroups. In: Discovery Science. pp. 51–66. Springer International Publishing (2016)
10. Proença, H.M., Grünwald, P., Bäck, T., van Leeuwen, M.: Robust subgroup discovery. Data Mining and Knowledge Discovery (2022)
11. Proença, H.M., Grünwald, P., Bäck, T., Leeuwen, M.v.: Discovering Outstanding Subgroup Lists for Numeric Targets Using MDL. In: Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020). pp. 19–35 (2021)
12. Semenova, L., Rudin, C., Parr, R.: On the existence of simpler machine learning models. In: ACM Conference on Fairness, Accountability, and Transparency. p. 1827–1858. FAccT '22, Association for Computing Machinery (2022)
13. Xin, R., Zhong, C., Chen, Z., Takagi, T., Seltzer, M.I., Rudin, C.: Exploring the whole rashomon set of sparse decision trees. ArXiv **abs/2209.08040** (2022)