Subgroup Discovery with Small and Alternative Feature Sets

JAKOB BACH, Karlsruhe Institute of Technology (KIT), Germany

Subgroup-discovery methods find interesting regions in a dataset. In this article, we analyze two constraint types to enhance the interpretability of subgroups: First, we make subgroup descriptions small by limiting the number of features used. Second, we propose the novel problem of finding alternative subgroup descriptions, which cover a similar set of data objects as a given subgroup but use different features. We describe how to integrate both constraint types into heuristic subgroup-discovery methods as well as a novel Satisfiability Modulo Theories (SMT) formulation, which enables a solver-based search for subgroups. Further, we prove $\mathcal{NP}$-hardness of optimization with either constraint type. Finally, we evaluate unconstrained and constrained subgroup discovery with 27 binary-classification datasets. We observe that heuristic search methods often yield high-quality subgroups fast, even with constraints.

CCS Concepts: • **Theory of computation → Mixed discrete-continuous optimization**; *Problems, reductions and completeness*; • **Computing methodologies → Feature selection**; *Supervised learning*.

Additional Key Words and Phrases: Subgroup Discovery, Constraints, Feature Selection, Alternatives, Satisfiability Modulo Theories, Explainability, Interpretability, XAI

ACM Reference Format:
Jakob Bach. 2025. Subgroup Discovery with Small and Alternative Feature Sets. *Proc. ACM Manag. Data* 3, 3 (SIGMOD), Article 221 (June 2025), 27 pages. https://doi.org/10.1145/3725358

1 Introduction

Motivation. Interpretable machine learning has gained importance in recent years [16, 62]. Some machine-learning models are simple enough to be intrinsically interpretable [16], e.g., subgroup descriptions. Subgroup discovery aims to identify 'interesting' subsets of a dataset [4], such as data objects sharing a specific class label, that can be described by concise conditions on feature values. Subgroup-discovery methods have recently been employed in various fields, such as chemistry [49], database engineering [71], decision making [80], medicine [22], and social sciences [39].

Figure 1 displays an exemplary rectangle-shaped subgroup description for a two-dimensional, real-valued dataset with a binary prediction target. This subgroup is defined by $(\mathit{Feature\_1} \in [3.0, 5.1]) \land (\mathit{Feature\_2} \in [1.0, 1.8])$ and contains a high share of data objects with $\mathit{Target} = 1$. If such subgroup descriptions become too complex, their interpretability suffers [59]. Thus, imposing constraints on subgroup descriptions may foster their interpretability.

Problem statement. This article addresses the problem of subgroup discovery with two types of constraints, which both relate to the features used in subgroup descriptions:

First, *feature-cardinality constraints* limit the number of selected, i.e., used, features. Thus, subgroup descriptions become *small*, which increases their interpretability for users at the potential expense of subgroup quality. For example, in Figure 1, a subgroup description may use either feature alone rather than both and still cover a high share of data objects with $\mathit{Target} = 1$. Feature selection is common for other machine-learning tasks as well [31, 50].

Author's Contact Information: Jakob Bach, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, jakob.bach@kit.edu.

ACM 2836-6573/2025/6-ART221
https://doi.org/10.1145/3725358

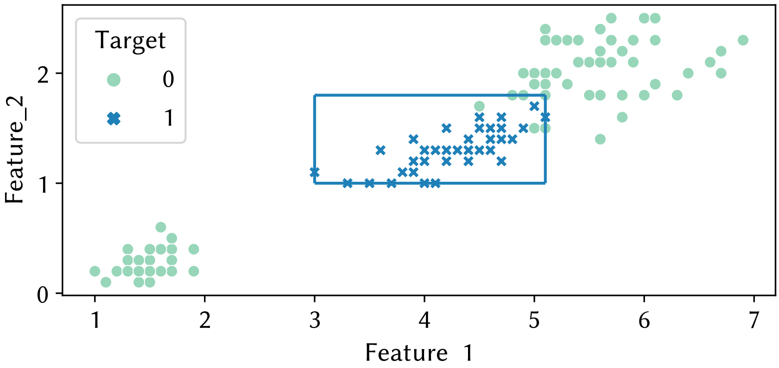


Fig. 1. Rectangular subgroup description for a dataset with two features and a binary prediction target.

Second, *alternative subgroup descriptions* should cover roughly the same set of data objects as a given subgroup but use different features. For instance, in Figure 1, one may define a subgroup based on one feature and then try to cover a similar set of data objects with bounds on the other feature. Such alternative subgroup descriptions provide alternative explanations for the same subgroup. Alternative explanations are also popular for other explainable-AI techniques like counterfactuals [63, 73], e.g., to enable users to develop and test several hypotheses or foster trust in the predictions [41, 78]. Consider subgroup discovery on scientific data, where a subgroup may capture samples exhibiting an interesting scientific phenomenon, and the subgroup description gives a possible explanation of which factors the phenomenon depends on. Thus, users may leverage the subgroup description to formulate a scientific hypothesis for the domain. Just considering one explanation may be misleading in such a situation. For example, different factors (features) may explain the same phenomenon (subgroup), particularly if factors are correlated or interact with each other. Thus, obtaining alternative subgroup descriptions may broaden the users' perspective and prevent selecting one explanation prematurely.

Related work. Widely used subgroup-discovery methods are algorithmic [4, 34], so considering particular constraint types on subgroup descriptions usually entails type-specific adaptations.

The number of features used in a subgroup description is an established measure for subgroup complexity [33, 34, 77]. However, systematic evaluations of this constraint type are lacking, as existing work typically only analyzes one subgroup-discovery method or one cardinality threshold.

Various subgroup-discovery methods yield a set of diverse subgroups [11, 15, 54, 76]. However, these alternative subgroups aim to cover different regions of the dataset rather than finding alternative descriptions for the same region. Approaches that explicitly target alternative descriptions [13, 25, 52, 76] are rarer and differ from our approach as well, as we discuss in Section 7.

Contributions. We make four main contributions:

(1) We formalize subgroup discovery as a Satisfiability Modulo Theories (SMT) optimization problem. This novel white-box formulation admits a solver-based search for subgroups and allows integrating and combining constraints in a declarative manner.

(2) We formalize feature-cardinality constraints and the novel notion of alternative subgroup descriptions. For the latter, we allow users to control the number and dissimilarity of alternatives. We also describe how to integrate both constraint types into three existing heuristic subgroup-discovery methods and two novel baselines.

(3) We show that finding optimal solutions under these two constraint types is computationally hard by proving $\mathcal{NP}$-completeness.

(4) We conduct comprehensive experiments with 27 binary-classification datasets from the Penn Machine Learning Benchmarks (PMLB) [66, 72] to compare solver-based and seven algorithmic search methods for subgroups, with and without constraints. We publish all code[1] and experimental data[2]. We observe that two heuristic search methods yield similar subgroup quality as solver-based search while being one to two orders of magnitude faster. Feature-cardinality constraints reduce overfitting, and a few features suffice to reach comparable subgroup quality as without constraints. Finally, there may be multiple alternative subgroup descriptions with comparable quality and similarity to the original subgroup description.

Outline. The remainder of this article is structured as follows: Section 2 introduces fundamentals. Section 3 proposes two baselines for subgroup discovery. Section 4 describes and analyzes constrained subgroup discovery. Section 5 outlines our experimental design. Section 6 presents the corresponding experimental results. Section 7 reviews related work. Section 8 concludes.

2 Fundamentals

In this section, we introduce relevant fundamentals of subgroup discovery: the optimization problem (Section 2.1) and common algorithmic search methods (Section 2.2).

2.1 Problem of Subgroup Discovery

To harmonize formalization and evaluation, we focus on real-valued datasets with a binary prediction target. $X \in \mathbb{R}^{m \times n}$ denotes a dataset in the form of a matrix. Each row is a data object, and each column is a feature. $y \in \{0, 1\}^m$ represents the binary prediction target. Section 6.3 discusses adaptations for other scenarios.

For such real-valued datasets, a subgroup description typically forms a hyperrectangle in $\mathbb{R}^n$ by defining a lower bound and an upper bound for each feature. The bounds may also be infinite to leave a feature unrestricted. A data object resides in the subgroup if all the data object's feature values satisfy the subgroup description's bounds:

Definition 1 (Subgroup (description)). Given a dataset $X \in \mathbb{R}^{m \times n}$, a *subgroup* is described by its lower bounds $lb \in \{\mathbb{R} \cup \{-\infty, +\infty\}\}^n$ and upper bounds $ub \in \{\mathbb{R} \cup \{-\infty, +\infty\}\}^n$. Data object $X_{i\cdot}$ is a *member* of this subgroup if $\forall j \in \{1, \dots, n\} : \left(X_{ij} \geq lb_j\right) \land \left(X_{ij} \leq ub_j\right)$.

To frame subgroup discovery as an optimization problem, let $Q(lb, ub, X, y)$ be the subgroup quality for a particular dataset:

Definition 2 (Subgroup discovery). Given a dataset $X \in \mathbb{R}^{m \times n}$ with prediction target $y \in \{0, 1\}^m$, *subgroup discovery* is the problem of finding a subgroup (Definition 1) with bounds $lb, ub \in \{\mathbb{R} \cup \{-\infty, +\infty\}\}^n$ that maximizes a given notion of subgroup quality $Q(lb, ub, X, y)$.

For binary prediction targets, high-quality subgroups should typically contain many data objects from one class but few from the other. Without loss of generality, we assume that the class with label '1' is the class of interest, also called *positive* class. Weighted Relative Accuracy (WRAcc) [44] is a popular metric for subgroup quality [59]:

$$\text{WRAcc} = \frac{m_b}{m} \cdot \left(\frac{m_b^+}{m_b} - \frac{m^+}{m} \right) = \frac{m_b^+}{m} - \frac{m_b \cdot m^+}{m^2} \tag{1}$$

[1] https://github.com/Jakob-Bach/Constrained-Subgroup-Discovery; additionally backed up in the *Software Heritage archive* with the identifier `swh:1:dir:e587f3e572f07d879abae884f4a0c8aea8b0887d`

[2] https://doi.org/10.35097/nftgaf7w73hy2491

Besides the total number of data objects m, this metric considers the number of positive data objects m^+, the number of subgroup members m_b, and the number of positive subgroup members m_b^+. In particular, WRAcc is the product of two factors, i.e., the relative frequency of subgroup membership and the prevalence of the positive class in the subgroup compared to the whole dataset.

The maximum value of WRAcc depends on the frequencies of the two classes in the dataset. Thus, it makes sense to normalize WRAcc when working with datasets with different class frequencies. One option is a max-normalization to the range $[-1, 1]$ [56]:

$$\text{nWRAcc} = \frac{\text{WRAcc}}{\text{WRAcc}_{\max}} = \frac{m_b^+ \cdot m - m^+ \cdot m_b}{m^+ \cdot (m - m^+)} \tag{2}$$

2.2 Subgroup-Discovery Methods

To discover subgroups, there are heuristic search methods, like PRIM [24] and Best Interval [55], as well as exhaustive search methods, like SD-Map [5, 6], MergeSD [28], and BSD [46, 48]. In this section, we discuss five search methods that are relevant for our experiments; see [4, 33, 34, 77] for comprehensive surveys of subgroup discovery.

Patient Rule Induction Method (PRIM). PRIM [24] consists of a peeling phase, which iteratively restricts the subgroup's bounds, and a pasting phase, which iteratively expands them. In this article, we only use the peeling phase since pasting may have little effect on subgroup quality and is often left out [2]. Peeling starts with a subgroup containing all data objects. Each iteration excludes a fraction $\alpha \in (0, 1)$ of data objects from the subgroup by setting a corresponding lower or upper bound on a feature. Having tested two new bounds for each feature, the algorithm uses the subgroup with the highest quality for the next iteration. Once the current subgroup contains at most a fraction $\beta_0 \in [0, 1]$ of data objects, *PRIM* returns the best subgroup from all iterations.

Beam Search (Beam). Beam search is a generic search strategy that is common in subgroup discovery as well [4]. It maintains a set of currently best solution candidates, whose number is the beam width $w \in \mathbb{N}$. *Beam* starts with w unrestricted subgroup descriptions. Each iteration tries to refine each solution candidate by updating either the lower or upper bound of one feature. In contrast to *PRIM*, all unique feature values in the subgroup may serve as new bounds. The new beam comprises the highest-quality w subgroups from the updates and the previous iteration. Once the beam remains unchanged, the best of its subgroups is returned.

Best Interval (BI). BI [55] is a subgroup-refinement procedure based on theoretical properties of the quality metric WRAcc (Equation 1). It can be used within a beam-search strategy. Unlike *Beam*, *BI* updates a feature's lower and upper bounds simultaneously rather than separately but still requires only one pass over the feature's values to find the best update option.

SD-Map. SD-Map [6] is an exhaustive search method based on the pattern-mining algorithm *FP-growth* [32]. It assumes that numeric features are discretized and produces equality conditions of the form $X_{ij} = v$ instead of the numeric intervals we focus on (Definition 1). The extension *SD-Map** [5] adds support for numeric targets and quality-based pruning of the search space.

Bitset-based Subgroup Discovery (BSD). BSD [48] is another exhaustive search method that produces equality conditions in the subgroup description. It combines a branch-and-bound strategy with a special binary data representation and pruning techniques to speed up the search. *NumBSD* [46] can handle numeric targets, though the features are still assumed to be discrete.

Algorithm 1: *Random Search* for subgroup discovery.

```
   Input: Dataset $X \in \mathbb{R}^{m \times n}$,
          Prediction target $y \in \{0,1\}^m$,
          Subgroup-quality function $Q(lb, ub, X, y)$,
          Number of iterations $n\_iters \in \mathbb{N}$
   Output: Subgroup bounds $lb, ub \in \{\mathbb{R} \cup \{-\infty, +\infty\}\}^n$
1  $Q^{\text{opt}} \leftarrow -\infty$
2  for $iters \leftarrow 1$ to $n\_iters$ do
3      for $j \leftarrow 1$ to $n$ do
4          $(lb_j,\ ub_j) \leftarrow (-\infty, +\infty)$
5      for $j \in$ get_permissible_feature_idxs(...) do
6          $(lb_j,\ ub_j) \leftarrow$ sample_uniformly(unique($X_{\cdot j}$))
7      if $Q(lb, ub, X, y) > Q^{opt}$ then
8          $Q^{\text{opt}} \leftarrow Q(lb, ub, X, y)$
9          $(lb^{\text{opt}},\ ub^{\text{opt}}) \leftarrow (lb,\ ub)$
10 for $j \leftarrow 1$ to $n$ do
11     if $lb_j^{opt} = \min_{i \in \{1,\ldots,m\}} X_{ij}$ then $lb_j^{\text{opt}} \leftarrow -\infty$
12     if $ub_j^{opt} = \max_{i \in \{1,\ldots,m\}} X_{ij}$ then $ub_j^{\text{opt}} \leftarrow +\infty$
13 return $lb^{opt},\ ub^{opt}$
```

3 Baselines and Preliminaries

In this section, we propose two *baselines* for subgroup discovery, *Random* (Section 3.1) and *MORS* (Section 3.2). They are conceptually simpler than the heuristic search methods from Section 2.2 and serve as additional reference points in our experiments. Section 3.2 also lays essential groundwork for our complexity analyses in Sections 4.2.5 and 4.3.5.

3.1 Random

The baseline *Random* (Algorithm 1) generates and evaluates subgroups for a fixed number of iterations $n\_iters \in \mathbb{N}$. In each iteration, this baseline randomly samples a lower bound and an upper bound for each feature, selecting these bounds uniformly from the feature's unique values (Lines 3–6). To support feature-cardinality constraints, which we discuss later (Section 4.2), the function *get_permissible_feature_idxs(...)* (Line 5) can restrict which features may be bounded; by default, all features are permissible. Further, bounds become infinite if they do not exclude any data objects from the subgroup (Lines 10–12). The algorithm tracks the best subgroup generated so far over the iterations (Lines 7–9) and finally returns it.

3.2 Minimal Optimal-Recall Subgroup (MORS)

This baseline builds on the following definition:

Definition 3 (Minimal Optimal-Recall Subgroup (MORS)). Given a dataset $X \in \mathbb{R}^{m \times n}$ with prediction target $y \in \{0,1\}^m$, the *Minimal Optimal-Recall Subgroup (MORS)* is the subgroup (Definition 1) whose lower and upper bounds of each feature correspond to the minimum and maximum value of that feature over all positive data objects (i.e., with $y_i = 1$) from the dataset X.

Algorithm 2: *MORS* for subgroup discovery.

```
Input: Dataset $X \in \mathbb{R}^{m \times n}$,
       Prediction target $y \in \{0,1\}^m$
Output: Subgroup bounds $lb, ub \in \{\mathbb{R} \cup \{-\infty, +\infty\}\}^n$
1 for $j \leftarrow 1$ to $n$ do
2     $lb_j \leftarrow \min_{\substack{i \in \{1,\dots,m\} \\ y_i = 1}} X_{ij}$
3     $ub_j \leftarrow \max_{\substack{i \in \{1,\dots,m\} \\ y_i = 1}} X_{ij}$
4     if $lb_j = \min_{i \in \{1,\dots,m\}} X_{ij}$ then $lb_j \leftarrow -\infty$
5     if $ub_j = \max_{i \in \{1,\dots,m\}} X_{ij}$ then $ub_j \leftarrow +\infty$
6 for $j \notin$ get_permissible_feature_idxs(...) do
7     $(lb_j,\ ub_j) \leftarrow (-\infty, +\infty)$
8 return $lb, ub$
```

The definition ensures that all positive data objects are subgroup members. Thus, the evaluation metric *recall*, i.e., the fraction of positive data objects becoming subgroup members, reaches its *optimal* value of 1. At the same time, raising the lower bounds or lowering the upper bounds would exclude positive data objects from the subgroup. In this sense, the set of subgroup members is *minimal*. The corresponding subgroup description is unique and solves the following variant of the subgroup-discovery problem:

Definition 4 (Minimal-optimal-recall-subgroup discovery). Given a dataset $X \in \mathbb{R}^{m \times n}$ with prediction target $y \in \{0,1\}^m$, *minimal-optimal-recall-subgroup discovery* is the problem of finding a subgroup (Definition 1) that contains as few negative data objects (i.e., with $y_i = 0$) as possible but all positive data objects (i.e., with $y_i = 1$) from the dataset X.

I.e., the problem aims to minimize the number of false positives subject to producing no false negatives. Equivalently, it maximizes the number of true negatives, i.e., negative data objects excluded from the subgroup, subject to producing no false negatives.

Algorithm 2 implements the baseline (Lines 2–3), followed by adaptations for feature-cardinality constraints (Lines 4–7). Since *MORS* only needs to iterate over all data objects and features once to determine the minimum and maximum feature values, its time complexity is $O(m \cdot n)$. This places minimal-optimal-recall-subgroup discovery in the complexity class $\mathcal{P}$:

Proposition 1 (Complexity for Definition 4). *The problem of minimal-optimal-recall-subgroup discovery (Definition 4) can be solved in $O(m \cdot n)$.*

Interestingly, a kind of inverted problem definition, the Maximum Box problem, is $\mathcal{NP}$-hard [20]. The latter problem maximizes the number of true positives, i.e., positive data objects in the subgroup, subject to producing no false positives, thereby retaining an optimal precision of 1.

For later proofs, we also need the following concept [58]:

Definition 5 (Perfect subgroup). Given a dataset $X \in \mathbb{R}^{m \times n}$ with prediction target $y \in \{0,1\}^m$, a *perfect subgroup* is a subgroup (Definition 1) that contains all positive data objects (i.e., with $y_i = 1$) but no negative data objects (i.e., with $y_i = 0$) from the dataset X.

Perfect subgroups reach the maximum possible WRAcc (Equation 1) for a dataset. Next, we define a corresponding search problem:

Definition 6 (Perfect-subgroup discovery). Given a dataset $X \in \mathbb{R}^{m \times n}$ with prediction target $y \in \{0,1\}^m$, *perfect-subgroup discovery* is the problem of finding a perfect subgroup (Definition 5) if it exists or determining that it does not exist.

This problem is easy to solve with the help of Algorithm 2. In particular, after *MORS* has found a subgroup in $O(m \cdot n)$, one only needs to check whether the subgroup contains any negative data objects. If it does not, the subgroup is perfect, otherwise no perfect subgroup exists. In particular, the *MORS* bounds cannot be tightened to exclude negative data objects without also excluding positives, thereby violating the perfection of the subgroup. Thus, we obtain the following result:

Proposition 2 (Complexity for Definition 6). *The problem of perfect-subgroup discovery (Definition 6) can be solved in $O(m \cdot n)$.*

4 Constrained Subgroup Discovery

In this section, we propose an SMT encoding of subgroup discovery (Section 4.1) and discuss feature-cardinality constraints (Section 4.2) as well as alternative subgroup descriptions (Section 4.3).

4.1 SMT Encoding of Subgroup Discovery

Encoding subgroup discovery as a white-box optimization problem enables a solver-based search for subgroups. Such a formulation supports integrating constraints declaratively, while algorithmic search methods need specific adaptations for constraints. Here, we propose a Satisfiability Modulo Theories (SMT) encoding. SMT generally allows expressions in first-order logic with particular interpretations, e.g., arrays, arithmetic, or bit vectors [10]. Here, we use linear real arithmetic (LRA). The mixture of logical and arithmetic expressions in SMT makes it relatively straightforward to express subgroup discovery and the two constraint types we analyze. In preliminary experiments, we also considered an encoding as a mixed-integer linear program, but the latter formulation required more decision variables and constraints, and optimization was slower.

We define decision variables $lb, ub \in \{\mathbb{R} \cup \{-\infty, +\infty\}\}^n$ for the bounds of the subgroup description (Definition 1). The upper bounds naturally need to be at least as high as the lower bounds:

$$\forall j \in \{1, \dots, n\} : \; lb_j \leq ub_j \tag{3}$$

Next, we introduce binary decision variables $b_i \in \{0,1\}^m$ for subgroup membership. A data object is a member of the subgroup if all its feature values are contained within the bounds (Definition 1):

$$\forall i \in \{1, \dots, m\} : \; b_i \leftrightarrow \bigwedge_{j \in \{1, \dots, n\}} \left(\left(X_{ij} \geq lb_j \right) \land \left(X_{ij} \leq ub_j \right) \right) \tag{4}$$

Finally, we encode WRAcc (Equation 1) as the objective function. m_b and m_b^+ depend on the decision variables b_i while m and m^+ are dataset-dependent constants. Thus, WRAcc is linear in b_i:

$$\max \quad Q_{\text{WRAcc}} = \frac{1}{m} \cdot \sum_{\substack{i \in \{1, \dots, m\} \\ y_i = 1}} b_i - \frac{m^+}{m^2} \cdot \sum_{i=1}^{m} b_i \tag{5}$$

Instead of introducing the decision variables b_i, one could also insert the Boolean expression on the right-hand side of Equation 4 into Equation 5 directly. In particular, lb_j and ub_j are the only decision variables strictly necessary for the optimization problem. However, we also use the variables b_i for encoding alternative subgroup descriptions later and therefore make them explicit.

4.2 Feature-Cardinality Constraints

In this section, we discuss feature-cardinality constraints for subgroup discovery. First, we motivate and define them (Section 4.2.1). Next, we describe how to integrate them into our SMT encoding (Section 4.2.2), heuristic search methods (Section 4.2.3), and baselines (Section 4.2.4). Finally, we analyze the time complexity of subgroup discovery with this constraint type (Section 4.2.5).

4.2.1 Concept. Complex subgroup descriptions may be hard to interpret [59]. Feature-cardinality constraints simplify subgroup descriptions by limiting the number of features used in them. In particular, we define feature selection for subgroups as follows:

Definition 7 (Feature selection in subgroups). Given a dataset $X \in \mathbb{R}^{m \times n}$ and a subgroup (Definition 1) with bounds $lb, ub \in \{\mathbb{R} \cup \{-\infty, +\infty\}\}^n$, Feature j is *selected* if the bounds exclude at least one data object of X from the subgroup, i.e., $\exists i \in \{1, \dots, m\} : \left(X_{ij} < lb_j\right) \lor \left(X_{ij} > ub_j\right)$.

The bounds of unselected features can be considered infinite, effectively removing them from the subgroup description. The *feature cardinality* of a subgroup description is the number of selected features, sometimes called the description's *length* in the literature [4, 33]. A feature-cardinality constraint imposes an upper bound on the number of selected features:

Definition 8 (Feature-cardinality constraint). Given a cardinality threshold $k \in \mathbb{N}$, a *feature-cardinality constraint* for a subgroup (Definition 1) requires the subgroup to have at most k features selected (Definition 7).

A subgroup-discovery method may also select less than k features if selecting more does not improve subgroup quality.

4.2.2 SMT Encoding. To encode whether a feature is selected or not, we introduce binary decision variables $s, s^{\text{lb}}, s^{\text{ub}} \in \{0, 1\}^n$. Based on Definition 7, a feature is selected if its minimum is less than the lower bound or its maximum is greater than the upper bound:

$$\begin{aligned}
\forall j \in \{1, \dots, n\} : \quad & s_j^{\text{lb}} \leftrightarrow \left(lb_j > \min_{i \in \{1, \dots, m\}} X_{ij}\right) \\
\forall j \in \{1, \dots, n\} : \quad & s_j^{\text{ub}} \leftrightarrow \left(ub_j < \max_{i \in \{1, \dots, m\}} X_{ij}\right) \\
\forall j \in \{1, \dots, n\} : \quad & s_j \leftrightarrow \left(s_j^{\text{lb}} \lor s_j^{\text{ub}}\right)
\end{aligned} \tag{6}$$

Minimum and maximum feature values are constants that can be determined before formulating the optimization problem.

Given the variables s_j (Equation 6), setting an upper bound on the number of selected features (Definition 8) is straightforward:

$$\sum_{j=1}^{n} s_j \leq k \tag{7}$$

Instead of defining the decision variables s_j, s_j^{lb}, and s_j^{ub}, one could also insert the corresponding expressions from Equation 6 into Equation 7 directly.

4.2.3 Integration into Heuristic Search Methods (Section 2.2). The feature-cardinality constraint (Equation 7) has the form $|F_s| \leq k$ for the feature set F_s induced by the selection decisions $s \in \{0, 1\}^n$. Thus, the constraint is *antimonotonic* [65] regarding the set of selected features. In particular, the empty feature set satisfies the constraint for any $k \geq 0$. If a set of selected features satisfies the constraint, all its subsets also satisfy it. Vice versa, if a feature set violates the constraint, all its supersets violate it as well.

The antimonotonicity property eases integrating the constraint into *Beam*, *BI*, and *PRIM*. These three methods iteratively enlarge the set of selected features. In particular, they start with unrestricted subgroup bounds, i.e., an empty feature set. Each iteration may either add bounds on one further feature, thereby increasing the feature-set size by one, or refine the bounds on an already selected feature, so the feature-set size remains constant. Features cannot be deselected, so the feature-set size is non-decreasing overall. Once k features are selected, we only allow bounds on these features to be refined. Due to antimonotonicity, all potential feature supersets are invalid anyway. In contrast, any feature may be bounded as long as fewer than k features are selected.

4.2.4 Integration into Baselines (Section 3). For *Random*, we sample k out of n features uniformly random without replacement. This step implements the function *get_permissible_feature_idxs(...)* in Line 5 of Algorithm 1. Next, we sample these features' bounds as usual. For *MORS*, we employ a univariate, quality-based heuristic for feature selection in Line 6 of Algorithm 2: For each feature, we determine the number of negative data objects in the subgroup if only this feature uses *MORS* bounds. We select the k features with the lowest number of false positives. This heuristic is equivalent to selecting the features with the highest WRAcc for univariate *MORS* bounds.

4.2.5 Computational Complexity. Without constraints, an exhaustive search for subgroups needs to evaluate $O(m^{2n})$ subgroup descriptions. In particular, each feature has $O(m)$ unique values, resulting in $O(m^2)$ relevant lower-upper-bound combinations that are combined over n features. Thus, the number of features impacts the search space exponentially, while the number of data objects has a polynomial impact. In practice, this expression only is an upper bound: Exhaustive search methods may employ quality-based pruning to not evaluate all solution candidates explicitly [4].

With a feature-cardinality constraint, there are $\binom{n}{k} \leq n^k$ feature sets of size k with $O(m^{2k})$ bound candidates each, resulting in $O(n^k \cdot m^{2k})$ candidate subgroup descriptions. This complexity term is exponential in the number of selected features k. However, for a fixed k, the term is polynomial in the dataset size $m \cdot n$. Thus, the problem is in the parameterized complexity class $\mathcal{XP}$ [19]:

Proposition 3 (Parameterized complexity for Definition 2 with 8). *The problem of subgroup discovery (Definition 2) with a feature-cardinality constraint (Definition 8) resides in the parameterized complexity class $\mathcal{XP}$ for the parameter k.*

Without a feature-cardinality constraint, $\mathcal{XP}$ membership holds for the parameter n instead of k. [13] showed that it is an $\mathcal{NP}$-hard problem to find a subgroup description minimizing feature cardinality while inducing exactly the same subgroup membership as a given subgroup description. We adapt this hardness result to optimizing subgroup quality under a feature-cardinality constraint. First, we tackle the search problem for perfect subgroups:

Proposition 4 (Complexity for Definition 6 with 8). *The problem of perfect-subgroup discovery (Definition 6) with a feature-cardinality constraint (Definition 8) is $\mathcal{NP}$-complete.*

Proof. Let I be an arbitrary instance of the decision problem Set Covering [38]. I consists of a set of elements $E = \{e_1, \dots, e_m\}$, a set of sets $\mathbb{S} = \{S_1, \dots, S_n\}$ with $E = \bigcup_{S \in \mathbb{S}} S$, and a cardinality $k \in \mathbb{N}$. Set Covering asks whether a subset $\mathbb{C} \subseteq \mathbb{S}$ exists with $|\mathbb{C}| \leq k$ and $E = \bigcup_{S \in \mathbb{C}} S$, i.e., containing each element from E in at least one set and consisting of at most k sets.

We transform I into an instance I' of the perfect-subgroup-discovery problem with a feature-cardinality constraint (Definitions 6 and 8). First, we keep the cardinality threshold $k \in \mathbb{N}$. Next, we define the dataset $X \in \{0,1\}^{(m+1) \times n}$ such that X_{ij} encodes $e_i \in S_j$, i.e., membership of Element i in Set j. The index $i = m + 1$ represents a *dummy element* that is not part of any set, so all its feature values are set to 0. Finally, we define the prediction target $y \in \{0,1\}^{m+1}$ as $y_{m+1} = 1$ and

$y_i = 0$ otherwise. y represents whether an element should *not* be covered by $\mathbb{C} \subseteq \mathbb{S}$. In particular, all elements from E should be covered but not the dummy element.

A perfect subgroup (Definition 5) exactly replicates the prediction target as subgroup membership. Hence, with our definition of X and y for problem instance I', a perfect subgroup only contains the data object representing the dummy element. As all feature values of this dummy data object are 0, the subgroup description only consists of the bounds $lb_j = ub_j = 0$ for selected features. Thus, the data objects excluded from the subgroup assume the value 1 for at least one selected feature. I.e., all elements from E are in a selected set, so the selected features represent a set cover $\mathbb{C}$.

In contrast, if no feature set of the desired size k can describe a perfect subgroup, then always at least one data object with $y_i = 0$ has to be in the subgroup. Thus, at least one element from E must not be in any selected set, so no valid set cover of size k exists.

Overall, a solution to the instance I' of the perfect-subgroup-discovery problem with a feature-cardinality constraint also solves the instance I of Set Covering. Since the latter problem is $\mathcal{NP}$-hard [38], the former is as well. To be more precise, our problem is $\mathcal{NP}$-complete since checking a solution only entails a polynomial cost of $O(m \cdot n)$, requiring one pass over the dataset to determine subgroup membership and feature selection. □

This hardness result contrasts with the polynomial runtime of unconstrained perfect-subgroup discovery (Proposition 2), which corresponds to a cardinality constraint with $k = n$.

Generalizing Proposition 4, the optimization problem of subgroup discovery with a feature-cardinality constraint is $\mathcal{NP}$-complete under a reasonable assumption regarding subgroup quality:

Proposition 5 (Complexity for Definition 2 with 8). *Assuming a subgroup-quality function $Q(lb, ub, X, y)$ for which only perfect subgroups (Definition 5) reach its maximal value, the problem of subgroup discovery (Definition 2) with a feature-cardinality constraint (Definition 8) is $\mathcal{NP}$-complete.*

Proof. Let I be an arbitrary instance of the perfect-subgroup-discovery problem with a feature-cardinality constraint (Definitions 6 and 8). We transform I into an instance I' of the subgroup-discovery problem (Definition 2) with a feature-cardinality constraint. In particular, we optimize a subgroup-quality function $Q(lb, ub, X, y)$ rather than searching for a perfect subgroup (Definition 5). The other problem inputs remain the same.

Based on the assumption on $Q(lb, ub, X, y)$ in Proposition 5, the solution for I' is a perfect subgroup if the latter exists. Thus, if the optimal subgroup for I' is not perfect, then a perfect subgroup does not exist. Checking whether a subgroup is perfect entails a cost of $O(m \cdot n)$. Overall, an algorithm for subgroup discovery with a feature-cardinality constraint solves perfect-subgroup discovery with a feature-cardinality constraint with negligible overhead. Since the latter problem is $\mathcal{NP}$-complete (Proposition 4) and the former resides in $\mathcal{NP}$, the former is also $\mathcal{NP}$-complete. □

WRAcc (Equation 1) satisfies the assumption of Proposition 5 since only perfect subgroups, i.e., with $m_b^+ = m_b = m^+$, yield the theoretical maximum WRAcc.

4.3 Alternative Subgroup Descriptions

In this section, we propose the optimization problem of discovering alternative subgroup descriptions. First, we motivate and formalize the problem (Section 4.3.1). Next, we describe how to phrase it within our SMT encoding (Section 4.3.2), heuristic search methods (Section 4.3.3), and baselines (Section 4.3.4). Finally, we analyze the time complexity of this problem (Section 4.3.5).

4.3.1 Concept. We assume to have an *original subgroup* given, which may originate from any subgroup-discovery method. In a nutshell, alternative subgroup descriptions should induce similar subgroup membership of data objects as the original subgroup but select different features. One can search multiple such alternative descriptions sequentially. In this case, alternative descriptions should also select different features than all previous alternatives. We count the original subgroup as the zeroth alternative. Formally, we define the optimization problem as follows:

Definition 9 (Alternative-subgroup-description discovery). Given

- a dataset $X \in \mathbb{R}^{m \times n}$,
- $a \in \mathbb{N}$ existing subgroups with subgroup membership $b^{(l)} \in \{0,1\}^m$ and feature selection $s^{(l)} \in \{0,1\}^n$ for $l \in \{0, \dots, a-1\}$,
- a similarity measure $\text{sim}(\cdot)$ for subgroup-membership vectors,
- a dissimilarity measure $\text{dis}(\cdot)$ for feature-selection vectors of subgroups,
- and a dissimilarity threshold $\tau \in \mathbb{R}_{\geq 0}$,

alternative-subgroup-description discovery is the problem of finding a subgroup (Definition 1) with membership $b^{(a)} \in \{0,1\}^m$ and feature selection $s^{(a)} \in \{0,1\}^n$ that maximizes the subgroup-membership similarity $\text{sim}(b^{(a)}, b^{(0)})$ to the original subgroup while being dissimilar to all existing subgroups regarding the feature selection, i.e., $\forall l \in \{0, \dots, a-1\}: \ \text{dis}(s^{(a)}, s^{(l)}) \geq \tau$.

Users can control the number of alternatives $a \in \mathbb{N}$ and the dissimilarity threshold $\tau \in \mathbb{R}_{\geq 0}$. Due to the sequential search procedure that yields alternatives one by one, users can stop the search after each alternative. Further, we recommend employing a feature-cardinality constraint (Definition 8) so subgroup descriptions are easier to interpret and sufficiently many features are left for alternatives. Next, we discuss our choice of $\text{sim}(\cdot)$ and $\text{dis}(\cdot)$.

Similarity in objective function. Various options exist to quantify the similarity between two binary subgroup-membership vectors $b', b'' \in \{0,1\}^m$. We turn the Hamming distance [17] into a similarity measure and normalize it to $[0,1]$ with the vector lengths, obtaining the following *normalized Hamming similarity*:

$$\text{sim}_{\text{nHamm}}(b', b'') = \frac{1}{m} \cdot \sum_{i=1}^{m} (b'_i = b''_i) \tag{8}$$

If either b' or b'' is constant, this similarity measure is linear in its remaining argument (Equation 11). Further, if one considers one vector to be a prediction and the other to be the ground truth, Equation 8 equals prediction accuracy for binary classification.

Another popular similarity measure for sets or binary vectors is the Jaccard index [17], which relates the overlap of positive vector entries to their union:

$$\text{sim}_{\text{Jacc}}(b', b'') = \frac{\sum_{i=1}^{m} (b'_i \land b''_i)}{\sum_{i=1}^{m} (b'_i \lor b''_i)} \tag{9}$$

However, this similarity measure is not linear in b' and b'', which limits its applicability in some white-box solvers. Thus, we opt for the normalized Hamming similarity as the objective function.

Dissimilarity in constraints. There are various options to quantify the dissimilarity between feature-selection vectors. We formulate a constraint with the following *deselection dissimilarity*:

$$\text{dis}_{\text{des}}(s^{\text{new}}, s^{\text{old}}) = \sum_{j=1}^{n} (\lnot s^{\text{new}}_j \land s^{\text{old}}_j) \geq \min\left(\tau_{\text{abs}},\ k^{\text{old}}\right) \tag{10}$$

This dissimilarity counts how many previously selected features are *not* selected in the new subgroup description, either being replaced by other features or the total number of selected features being reduced. A valid solution must deselect at least $\tau_{\text{abs}} \in \mathbb{N}$ features unless the number selected before (k^{old}) is smaller, in which case all these must be deselected. For maximum dissimilarity, none of the previously selected features may be selected again. This dissimilarity measure is asymmetric, i.e., $\text{dis}_{\text{des}}(s^{\text{new}}, s^{\text{old}}) \neq \text{dis}_{\text{des}}(s^{\text{old}}, s^{\text{new}})$. However, 'old' and 'new' are well-defined in sequential search for alternatives. We explicitly designed our deselection dissimilarity to satisfy two desirable properties, which common dissimilarity measures like the Jaccard distance or the Dice dissimilarity [17] violate: (1) If s^{old} is constant, Equation 10 is linear in s^{new}, even if the exact number of newly selected features is unknown yet. This property is useful for solver-based search. (2) Equation 10 is antimonotonic in the new feature selection, which is useful for heuristic search.

4.3.2 SMT Encoding. Reformulating Equation 8, we see that the objective is linear regarding the alternative membership vector $b^{(a)}$:

$$\begin{aligned}\text{sim}_{\text{nHamm}}(b^{(a)}, b^{(0)}) &= \frac{1}{m} \cdot \sum_{i=1}^{m} \left(b_i^{(a)} \leftrightarrow b_i^{(0)} \right) \\ &= \frac{1}{m} \cdot \Big(\sum_{\substack{i \in \{1,\ldots,m\} \\ b_i^{(0)}=1}} b_i^{(a)} + \sum_{\substack{i \in \{1,\ldots,m\} \\ b_i^{(0)}=0}} \neg b_i^{(a)} \Big)\end{aligned} \tag{11}$$

In particular, since $b^{(0)}$ is known and therefore constant, we employ the expression from the second line, involving two plain sums. The negation $\neg b_i^{(a)}$ may also be expressed as $1 - b_i^{(a)}$.

To formulate the dissimilarity constraints, we leverage that the feature-selection vector $s^{(l)}$ and the corresponding number of selected features $k^{(l)}$ are known for all existing subgroups as well. Thus, we instantiate and adapt Equation 10 as follows:

$$\forall l \in \{0, \ldots, a-1\} : \ \text{dis}_{\text{des}}(s^{(a)}, s^{(l)}) = \sum_{\substack{j \in \{1,\ldots,n\} \\ s_j^{(l)}=1}} \neg s_j^{(a)} \geq \min\left(\tau_{\text{abs}},\ k^{(l)}\right) \tag{12}$$

In particular, we only sum over features that were selected in the l-th existing subgroup and check whether they are deselected now.

4.3.3 Integration into Heuristic Search Methods (Section 2.2). The situation resembles integrating a feature-cardinality constraint (Section 4.2.3). In particular, since the dissimilarity constraint (Equation 10) is antimonotonic in the subgroup's feature selection, the constraint is suitable for heuristic search methods that iteratively enlarge the set of selected features. In each iteration, we only need to check how many previously selected features are selected again. In particular, once $k^{(l)} - \tau_{\text{abs}}$ features from an existing subgroup description with $k^{(l)}$ features are selected, no additional features from this subgroup may be selected for defining bounds.

4.3.4 Integration into Baselines (Section 3). Adapting our two baselines to alternative subgroup descriptions is not straightforward since the optimization objective changes, and the search space under the dissimilarity constraint (Equation 10) is more complex. Thus, we did not implement and evaluate concrete adaptations. In particular, *MORS* is tailored towards a particular notion of subgroup quality, i.e., recall, instead of optimizing subgroup-membership similarity. For *Random*, we would like to uniformly sample from a constrained search space, which is a hard problem in general [21]. We could also sample from the unconstrained space until a valid feature set is found, which may take a long time. Another option is non-uniform sampling, e.g., only sample features not selected in any existing subgroup, which ignores the permitted feature-set overlap.

4.3.5 Computational Complexity. The size of the search space for alternative subgroup descriptions is comparable to that of discovering original subgroups, i.e., exponential in the number of features. The main difference is that the dissimilarity constraint needs to be considered. We prove $\mathcal{NP}$-completeness for a special case of alternative-subgroup-description discovery (Definition 9) first. To this end, we introduce the following definition:

Definition 10 (Perfect alternative subgroup description). Given

- a dataset $X \in \mathbb{R}^{m \times n}$,
- an original subgroup with subgroup membership $b^{(0)} \in \{0,1\}^m$ and feature selection $s^{(0)} \in \{0,1\}^n$,
- a dissimilarity measure $\text{dis}(\cdot)$ for feature-selection vectors of subgroups,
- and a dissimilarity threshold $\tau \in \mathbb{R}_{\geq 0}$,

a *perfect alternative subgroup description* defines a subgroup (Definition 1) with membership $b^{(a)} \in \{0,1\}^m$ and feature selection $s^{(a)} \in \{0,1\}^n$ that exactly replicates the subgroup membership of the original subgroup, i.e., $b^{(a)} = b^{(0)}$, while being dissimilar regarding the feature selection, i.e., $\text{dis}(s^{(a)}, s^{(0)}) \geq \tau$.

In particular, the value of the subgroup-membership similarity is fixed rather than an optimization objective. Similar to perfect subgroups (Definition 5), perfect alternative subgroup descriptions only exist in some datasets. We now define a corresponding search problem:

Definition 11 (Perfect-alternative-subgroup-description discovery). Given

- a dataset $X \in \mathbb{R}^{m \times n}$,
- an original subgroup with subgroup membership $b^{(0)} \in \{0,1\}^m$ and feature selection $s^{(0)} \in \{0,1\}^n$,
- a dissimilarity measure $\text{dis}(\cdot)$ for feature-selection vectors of subgroups,
- and a dissimilarity threshold $\tau \in \mathbb{R}_{\geq 0}$,

perfect-alternative-subgroup-description discovery is the problem of finding a perfect alternative subgroup description (Definition 10) if it exists or determining that it does not exist.

This search problem is $\mathcal{NP}$-complete under a reasonable assumption on the employed notion of feature-selection dissimilarity:

Proposition 6 (Complexity for Definition 11 with 8). *Assuming a combination of a dissimilarity measure dis$(\cdot)$ and a dissimilarity threshold $\tau \in \mathbb{R}_{\geq 0}$ that prevents selecting any selected feature from the original subgroup description again, the problem of perfect-alternative-subgroup-description discovery (Definition 11) with a feature-cardinality constraint (Definition 8) is $\mathcal{NP}$-complete.*

Our deselection dissimilarity (Equation 10) as $\text{dis}(\cdot)$ satisfies the assumption from Proposition 6 if we choose a dissimilarity threshold $\tau_{\text{abs}} \geq k^{\text{old}}$, as in the following proof. Other dissimilarity measures should typically also have a threshold value that enforces zero overlap between feature sets. Additionally, the problem naturally remains $\mathcal{NP}$-complete when dropping the assumption.

Proof. Let I be an arbitrary instance of the perfect-subgroup-discovery problem with a feature-cardinality constraint (Definitions 6 and 8). We transform I into an instance I' of the perfect-alternative-subgroup-description-discovery problem (Definition 11), also with a feature-cardinality constraint. First, we retain the feature-cardinality threshold $k \in \mathbb{N}$. Next, we define the original subgroup for I' to be perfect (Definition 5), i.e., having subgroup membership $b^{(0)} = y$. Further, we choose the deselection dissimilarity (Equation 10) as $\text{dis}(\cdot)$ and set $\tau_{\text{abs}} = k$. In particular, no feature from the original subgroup description may be selected again. We define dataset $X' \in \mathbb{R}^{m \times (n+k)}$ of problem instance I' as dataset $X \in \mathbb{R}^{m \times n}$ of problem instance I with k extra features that were all selected in the original subgroup description, i.e., $\forall j \in \{n+1, \dots, n+k\} : s_j^{(0)} = 1$ and $\forall j \in \{1, \dots, n\} : s_j^{(0)} = 0$. For solving I', we need not explicitly define the k extra features but could imagine them to equal the prediction target y.

A solution for problem instance I' also solves I. In particular, the perfect alternative subgroup description (Definition 10) defines a perfect subgroup here since it perfectly replicates the original subgroup membership, which is perfect, i.e., $b^{(a)} = b^{(0)} = y$. Due to the dissimilarity constraint, the alternative subgroup description only selects features from the original dataset X. Thus, if a perfect alternative subgroup description for I' exists, it also solves I. If it does not exist, then there also is no other perfect subgroup for I.

Overall, an algorithm for perfect-alternative-subgroup-description discovery with a feature-cardinality constraint also solves perfect-subgroup discovery with a feature-cardinality constraint with negligible overhead. Since the latter problem is $\mathcal{NP}$-complete (Proposition 4) and evaluating a solution for the former problem entails a polynomial cost, the former is $\mathcal{NP}$-complete as well. □

Next, we switch from the search problem for perfect alternatives to the optimization problem for alternative subgroup descriptions in general. We establish $\mathcal{NP}$-completeness under a reasonable assumption on the employed notion of subgroup-membership similarity:

Proposition 7 (Complexity for Definition 9 with 8). *Assuming*

- *a combination of a dissimilarity measure dis$(\cdot)$ and a dissimilarity threshold $\tau \in \mathbb{R}_{\geq 0}$ that prevents selecting any selected feature from the original subgroup description again,*
- *and a similarity measure sim$(\cdot)$ for which only perfect alternative subgroup descriptions (Definition 10) reach its maximal value regarding the original subgroup,*

the problem of alternative-subgroup-description discovery (Definition 9) with a feature-cardinality constraint (Definition 8) is $\mathcal{NP}$-complete.

Proof. Let I be an arbitrary instance of the perfect-alternative-subgroup-description-discovery problem with a feature-cardinality constraint (Definitions 11 and 8). We transform I into a problem instance I' of the alternative-subgroup-description-discovery problem (Definition 9), also with a feature-cardinality constraint. In particular, we optimize subgroup-membership similarity sim$(\cdot)$ rather than searching for a perfect alternative subgroup description (Definition 10). The other problem inputs remain the same.

Based on the assumption we made on sim$(\cdot)$ in Proposition 7, the optimal solution for I' is a perfect alternative subgroup description if the latter exists. Thus, if the optimal subgroup description for I' is not a perfect alternative, then the latter does not exist. Overall, an algorithm for alternative-subgroup-description discovery with a feature-cardinality constraint solves perfect-alternative-subgroup-description discovery with a feature-cardinality constraint with negligible overhead. Since the latter problem is $\mathcal{NP}$-complete (Proposition 6) and the former problem resides in $\mathcal{NP}$, the former is $\mathcal{NP}$-complete as well. □

Normalized Hamming similarity (Equation 8) satisfies the sim$(\cdot)$ assumption from Proposition 7 since only perfect alternative subgroup descriptions yield the theoretical maximum similarity of 1.

5 Experimental Design

In this section, we introduce our experimental design: subgroup-discovery methods, experimental scenarios, evaluation metrics, datasets, and the implementation.

Subgroup-discovery methods. We compare eight search methods for subgroup discovery: three exhaustive ones and five heuristic ones.

As the exhaustive search method *SMT*, we use the solver *Z3* [12, 18] to tackle our SMT encoding (Section 4.1). We set solver timeouts to keep runtime under control. The solver returns the optimal solution if found, otherwise the best solution so far. The actual runtime is higher than the timeout due to the initialization effort before optimization. We evaluate twelve exponentially scaled timeout values, i.e., {1 s, 2 s, 4 s, . . . , 2048 s}. The default value is 2048 s unless specified otherwise.

We also evaluate two exhaustive search methods from related work, i.e., *BSD* [48] and *SD-Map* [6], based on their implementation in the package *SD4Py* [35]. Both methods require discretizing numeric features. We use their built-in equal-width discretization and tune it by choosing the best subgroup quality out of ten bin counts. i.e., $\{2, 3, 4, 5, 10, 15, 20, 30, 40, 50\}$.

As heuristics, we employ three search methods from related work (Section 2.2) and our two baselines (Section 3). For *PRIM*, we set the peeling fraction to $\alpha = 0.05$, consistent with other implementations [2, 43] and the recommended value range proposed by its authors [24]. We set the support threshold to $\beta_0 = 0$, so there is no constraint on subgroup size. For *Beam* and *BI*, we choose a beam width of $w = 10$, falling between default values used in other implementations [2, 47]. For *Random*, we set the number of iterations to $n\_iters = 1000$. *MORS* is parameter-free.

Experimental scenarios. First, we analyze feature-cardinality constraints (Section 4.2) for all eight subgroup-discovery methods. We evaluate $k \in \{1, 2, 3, 4, 5\}$ as upper bounds on the number of selected features and also compare to the unconstrained problem. Since the exhaustive search methods *BSD* and *SD-Map* did not finish within two days in the unconstrained setting for five datasets, we replace these missing values with the results for $k = 5$.

Second, we study alternative subgroup descriptions (Section 4.3) for *SMT* and *Beam*, i.e., the solver-based and one heuristic search method. We choose a small $k = 3$ to leave enough unused features for describing alternatives. We search for $a = 5$ alternative subgroup descriptions with a dissimilarity threshold $\tau_{\text{abs}} \in \{1, 2, 3\}$.

Evaluation metrics. We use three types of evaluation metrics:

(1) We quantify *subgroup quality* with *nWRAcc* (Equation 2). We conduct a stratified five-fold cross-validation. In particular, each run of a subgroup-discovery method uses only 80% of the data objects of a dataset as the training set, while the rest serves as the test set. Thus, we can investigate how well a subgroup description from the training set generalizes to data objects from the test set.

(2) We measure the *runtime* of the subgroup-discovery methods.

(3) To assess *subgroup similarity* of alternative subgroup descriptions, we use *normalized Hamming similarity* (Equation 8) and *Jaccard similarity* (Equation 9) for subgroup membership.

Datasets. We use binary-classification datasets from the Penn Machine Learning Benchmarks (PMLB) [66, 72]. In each dataset, we choose the less frequent class as the positive class. To avoid scenarios that may be too easy or lack enough features for alternative subgroup descriptions, we exclude datasets with fewer than 100 data objects or fewer than 20 features. Next, we exclude one dataset with 1000 features, which has significantly more features than all remaining datasets. Finally, we manually exclude datasets that are duplicates of others. In the end, we obtain 27 datasets with 106 to 9822 data objects and 20 to 168 features (Table 1). The datasets do not contain missing values, and categorical features are ordinally encoded by default.

Implementation. We implemented our experiments in Python 3.8. All code is available online (Section 1). We organized the subgroup-discovery methods and evaluation metrics as a Python package to ease reuse. Our experimental pipeline parallelizes over datasets, cross-validation folds, and subgroup-discovery methods, while each of these experimental tasks runs single-threaded. Executing the pipeline took about 34 hours on a server with 160 GB RAM and an *AMD EPYC 7551* CPU with 32 cores and 2.0 GHz base clock.

Table 1. Datasets from PMLB used in our experiments. m denotes the number of data objects and n the number of features.

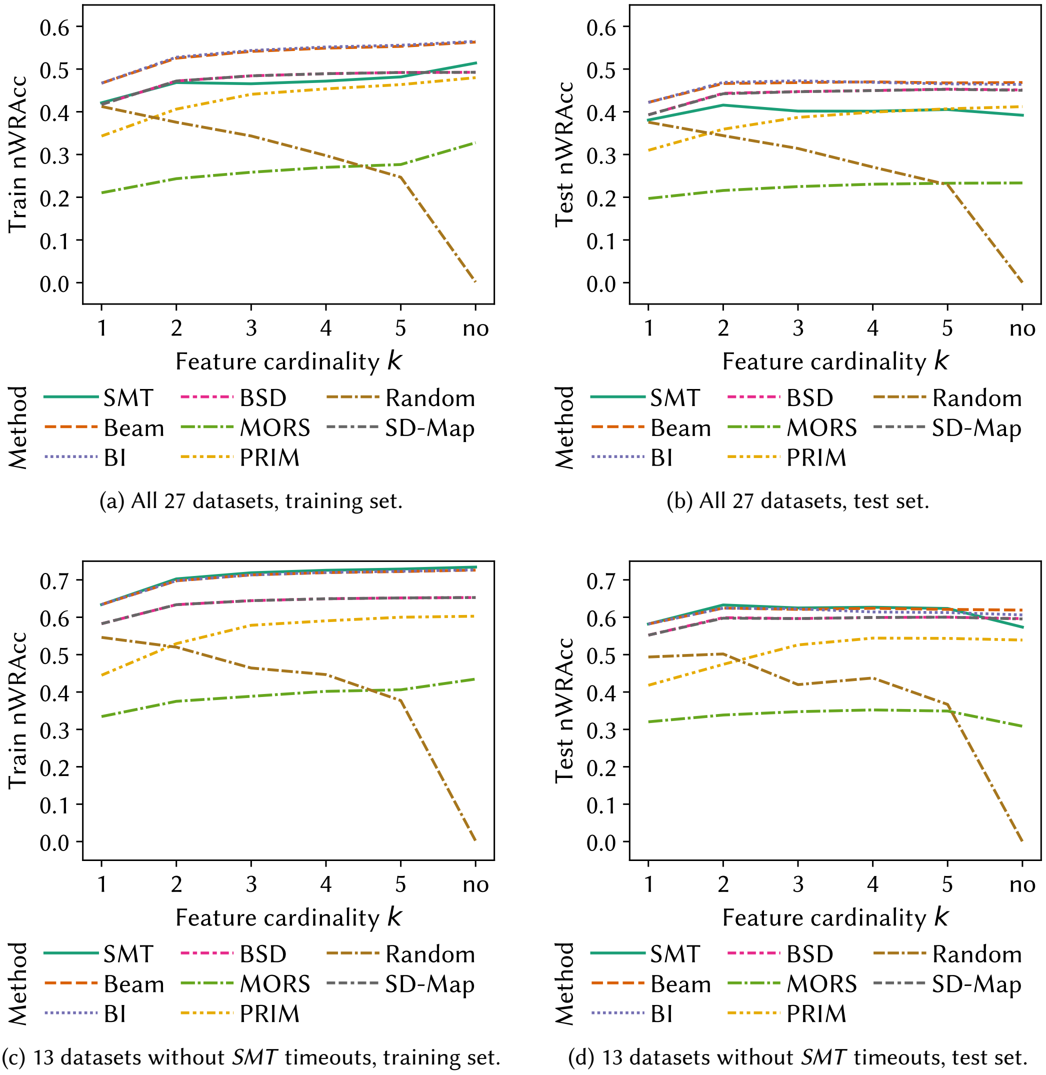
Dataset	m	n
backache	180	32
chess	3196	36
churn	5000	20
clean1	476	168
clean2	6598	168
coil2000	9822	85
credit_g	1000	20
dis	3772	29
GAMETES_Epistasis_2_Way_20atts_0.1H_EDM_1_1	1600	20
GAMETES_Epistasis_2_Way_20atts_0.4H_EDM_1_1	1600	20
GAMETES_Epistasis_3_Way_20atts_0.2H_EDM_1_1	1600	20
GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_50_EDM_2_001	1600	20
GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_75_EDM_2_001	1600	20
Hill_Valley_with_noise	1212	100
horse_colic	368	22
hypothyroid	3163	25
ionosphere	351	34
molecular_biology_promoters	106	57
mushroom	8124	22
ring	7400	20
sonar	208	60
spambase	4601	57
spect	267	22
spectf	349	44
tokyo1	959	44
twonorm	7400	20
wdbc	569	30

6 Evaluation

We evaluate feature-cardinality constraints first (Section 6.1) and alternative subgroup descriptions second (Section 6.2). Finally, we summarize and discuss the experimental results (Section 6.3).

6.1 Feature-Cardinality Constraints

Training-set subgroup quality. Figure 2a displays the training-set nWRAcc for subgroup-discovery methods and feature-cardinality thresholds k, averaged over datasets and cross-validation folds. The heuristic search methods *Beam* and *BI* are best on average and even outperform the solver-based search method *SMT*. This outcome is possible because *SMT* may run into timeouts and then yield suboptimal results. Figure 3a visualizes how many of the *SMT* optimization tasks for original subgroups finished within the evaluated solver timeouts. Additionally, Figure 3b shows how the mean subgroup quality of *SMT* increases with higher solver-timeout settings. For the maximum timeout setting of 2048 s and without a feature-cardinality constraint, 67.4% of the *SMT* searches finished, and 17 out of 27 datasets did not encounter timeouts. However, even if we limit our



(a) All 27 datasets, training set.

(b) All 27 datasets, test set.

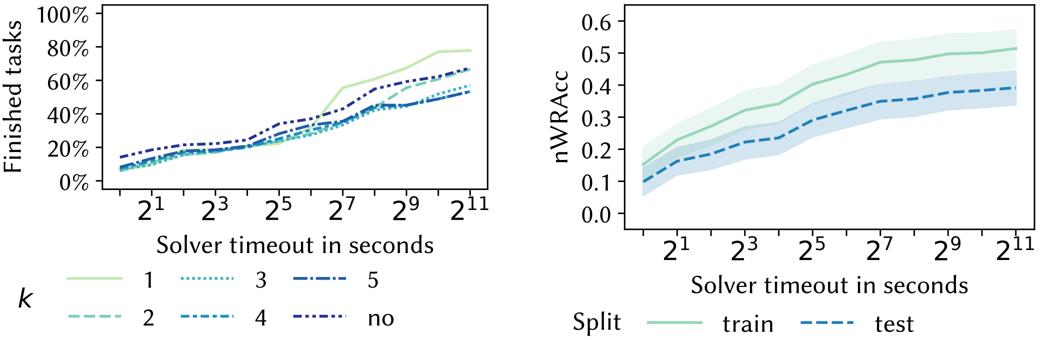
(c) 13 datasets without *SMT* timeouts, training set.

(d) 13 datasets without *SMT* timeouts, test set.

Fig. 2. Mean subgroup quality over datasets and cross-validation folds, by subgroup-discovery method and feature cardinality k. Results from the search for original subgroups.

evaluation to the datasets without solver timeouts (Figure 2c), *Beam* and *BI* are still remarkably close to the optimum quality. Note that this result is not specific to our method *SMT* but highlights a strong competition for any other exhaustive search method.

BSD and *SD-Map* yield worse average subgroup quality than *Beam* and *BI* as well. While the former two theoretically are exhaustive, they require discretization of numeric features. Thus, they effectively only have a fixed set of intervals to use as bounds instead of being able to choose the bounds independently at each possible feature value. *PRIM*'s worse subgroup quality may equally arise from a reduced search space. Although it follows an iterative subgroup-refinement procedure like *Beam* and *BI*, its refinement options are more limited. In particular, *PRIM* always has to remove a fixed fraction α of data objects from the subgroup, while *Beam* and *BI* can remove more or less.



(a) Frequency of finished *SMT* tasks over datasets and cross-validation folds, by feature cardinality k.

(b) Mean subgroup quality, with 95% confidence intervals based on datasets and cross-validation folds. Results without a feature-cardinality constraint.

Fig. 3. Impact of solver timeouts for *SMT* as the subgroup-discovery method. Results from the search for original subgroups.

Finally, the two baselines *MORS* and *Random* usually perform worse than the heuristic search methods, as one could expect due to their simplicity.

Test-set subgroup quality and overfitting. The trends on the test set are similar (Figures 2b and 2d). Further, the difference between training-set and test-set nWRAcc is higher for *SMT* than for the heuristic search methods. In other words, solutions from exhaustive search tend to overfit, i.e., generalize less. For example, the average difference between training-set nWRAcc and test-set nWRAcc without a feature-cardinality constraint is 0.122 for *SMT*, 0.101 for *BI*, 0.095 for *Beam*, 0.094 for *MORS*, 0.068 for *PRIM*, 0.042 for *SD-Map*, *0.041* for *BSD*, and 0.001 for *Random*. In this unconstrained scenario, *Beam* and *BI* have a higher mean test-set nWRAcc than *SMT* even on the datasets without solver timeouts. The relatively low overfitting of the exhaustive search methods *BSD* and *SD-Map* may be explained by their reduced search space due to feature discretization.

Impact of k on subgroup quality. Figure 2a also shows that the mean training-set nWRAcc increases with k for most subgroup-discovery methods, though the marginal utility decreases. In particular, even with $k = 1$, the mean nWRAcc is clearly above 50% of the nWRAcc achieved without a feature-cardinality constraint. Further, the quality increase is typically greatest between $k = 1$ and $k = 2$. These results indicate that small subgroup descriptions, which tend to be more interpretable, are already of high subgroup quality. *PRIM* exhibits a larger increase of subgroup quality with higher k than *Beam* and *BI*, narrowing the quality gap to the latter. *Random* differs from the other subgroup-discovery methods since its subgroup quality clearly decreases over k. This behavior occurs because *Random* samples bounds independently for each feature (Line 6 in Algorithm 1). As more features are used in the subgroup description, the expected number of data objects in the subgroup decreases. This reduction in subgroup size negatively impacts nWRAcc.

On the test set (Figure 2b), the benefit of a higher k is even smaller. For instance, the mean test-set nWRAcc of all methods except *PRIM* barely improves beyond $k = 2$. One reason is that higher feature-cardinality thresholds k exhibit more overfitting for all subgroup-discovery methods. E.g., the mean difference between training-set and test-set nWRAcc for *Beam* is 0.045 for $k = 1$, 0.073 for $k = 3$, and 0.095 without setting k. From the eight subgroup-discovery methods, *BSD* and *SD-Map* show the smallest increase of overfitting with higher k, *MORS* and *SMT* the largest.

Table 2. Mean runtime (in seconds) over datasets and cross-validation folds, by subgroup-discovery method and feature cardinality k. Results from the search for original subgroups.

k	1	2	3	4	5	no
BI	7.8	11.7	14.2	16.7	18.7	35.0
BSD	0.9	0.9	0.9	2.7	29.5	55.7
Beam	6.8	10.1	12.8	14.6	16.1	30.5
MORS	0.0	0.0	0.0	0.0	0.0	0.0
PRIM	0.1	0.2	0.3	0.3	0.5	1.3
Random	0.6	0.6	0.6	0.7	0.7	0.9
SD-Map	2.3	3.3	9.6	54.0	345.2	367.4
SMT	648.2	911.3	1091.7	1113.4	1117.4	849.0

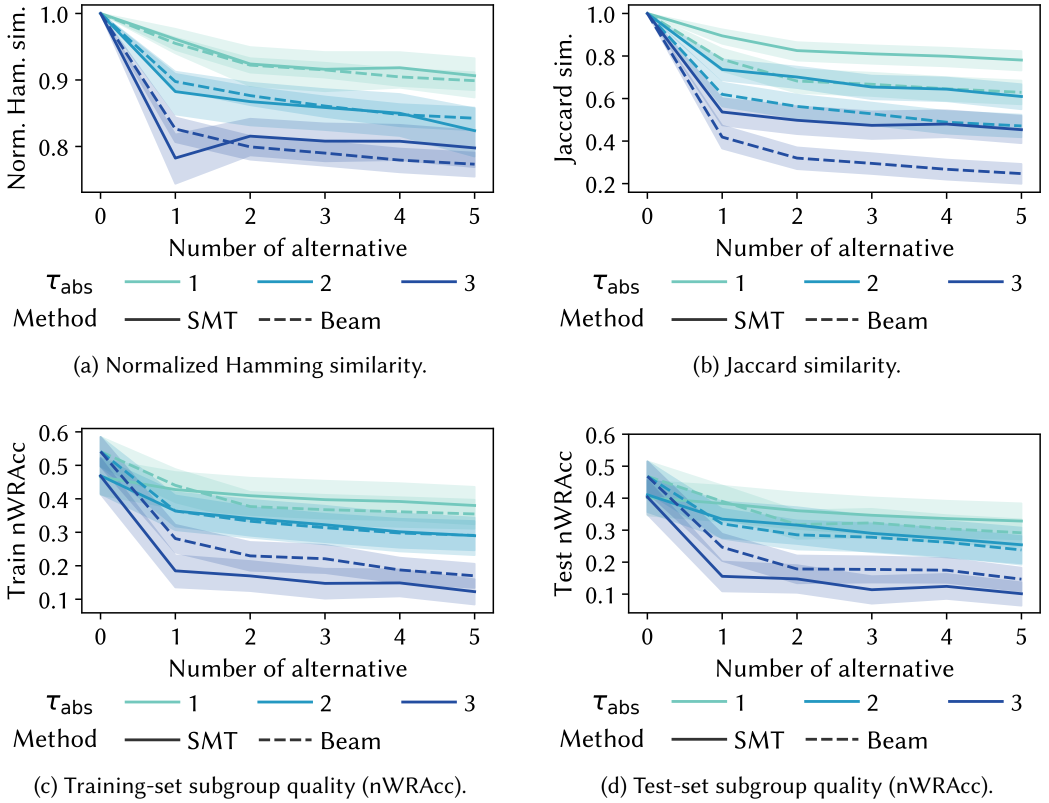
Table 3. Spearman correlation between runtime and metrics for dataset size, over datasets and cross-validation folds, by subgroup-discovery method. Results from the search for original subgroups without a feature-cardinality constraint, only using the 17 datasets without *SMT* timeouts.

Method	Σn^u	$m \cdot n$	m	n
BI	0.95	0.51	0.32	0.67
BSD	0.46	0.60	0.44	0.42
Beam	0.96	0.49	0.30	0.66
MORS	0.27	0.57	0.51	0.26
PRIM	0.84	0.56	0.29	0.76
Random	0.58	0.69	0.42	0.77
SD-Map	0.43	0.65	0.47	0.45
SMT	0.39	0.73	0.70	0.23

Runtime. Table 2 displays the mean runtime of the subgroup-discovery methods over k. The solver-based search method *SMT* is one to two orders of magnitude slower than the heuristic search methods *Beam* and *BI* and the exhaustive search methods with discretization, i.e., *BSD* and *SD-Map*. The heuristic *PRIM* and the baseline *Random* are yet another one to two orders of magnitude faster. Finally, the baseline *MORS* finishes in a fraction of a second, making it ideal for quickly obtaining a rough lower bound on subgroup quality. Overall, the heuristic search methods provide the best combination of subgroup quality and runtime. Among the three heuristics, *PRIM* offers the fastest runtime but the lowest subgroup quality. Thus, users should consider this trade-off between speed and quality when choosing a method.

As Table 2 shows as well, the heuristic search methods as well as *BSD* and *SD-Map* become faster with lower k. The trend also applies to the baseline *Random*, although to a lesser extent. *MORS* finishes instantly in all cases. In contrast, the picture for the solver-based search method *SMT* is less clear. Its mean runtime increases from $k = 1$ to $k = 3$ but remains roughly stable for $k \in \{4, 5\}$ and decreases without the feature-cardinality constraint, only remaining higher than for $k = 1$.

To determine which factors influence runtime besides k, we analyze the Spearman correlation between runtime and four simple metrics for dataset size. In particular, Table 3 considers the number of data objects m, the number of features n, the product of these two quantities $m \cdot n$, and the number of unique values per feature summed over the features Σn^u. For the three heuristic



(a) Normalized Hamming similarity.

(b) Jaccard similarity.

(c) Training-set subgroup quality (nWRAcc).

(d) Test-set subgroup quality (nWRAcc).

Fig. 4. Mean similarity and quality of alternative subgroup descriptions, with 95% confidence intervals based on datasets and cross-validation folds, by subgroup-discovery method, number of alternative, and dissimilarity threshold τ_{abs}.

search methods, the latter metric shows a high correlation to runtime, while the three exhaustive search methods exhibit the highest runtime correlation to $m \cdot n$.

6.2 Alternative Subgroup Descriptions

Subgroup similarity. Figures 4a and 4b visualize the average similarity between the original subgroup and the subgroups induced by alternative subgroup descriptions. As one would expect, subgroup-membership similarity decreases for more alternatives and the more the subgroup descriptions should differ. Further, the decrease is strongest from the original subgroup, i.e., the zeroth alternative, to the first alternative but then becomes smaller. This observation indicates that one may find several alternative subgroup descriptions of comparable similarity to the original one. These trends hold for the normalized Hamming similarity (Equation 8 and Figure 4a) as well as the Jaccard similarity (Equation 9 and Figure 4b). The latter yields lower similarity values than the former since it ignores data objects not contained in either of the compared subgroups.

Further, the observed trends exist for the solver-based search method *SMT* as well as the heuristic *Beam*. *SMT* yields more similar alternative descriptions than *Beam* for the Jaccard similarity, while the normalized Hamming similarity does not show a clear winner.

Table 4. Mean runtime (in seconds) over datasets and cross-validation folds, by subgroup-discovery method, dissimilarity threshold τ_{abs}, and number of alternative. Results from the search for alternative subgroup descriptions.

Method	τ_{abs}	Number of alternative					
		0	1	2	3	4	5
Beam	1	12.8	8.0	7.6	7.3	7.3	7.3
	2	12.8	7.7	7.4	7.2	7.0	6.8
	3	12.8	5.8	5.1	4.7	4.1	3.5
SMT	1	1091.7	166.0	221.5	239.6	258.1	277.9
	2	1105.2	377.5	463.5	537.5	599.4	658.3
	3	1107.4	869.1	670.8	597.6	588.1	557.6

Subgroup quality. The average subgroup quality of alternative subgroup descriptions (Figures 4c and 4d) shows similar trends as subgroup similarity, on the training set as well as on the set test. In particular, quality also decreases over the dissimilarity threshold τ_{abs} and over the number of alternatives a, with the largest decrease to the first alternative. For the highest dissimilarity threshold $\tau_{\text{abs}} = 3$, *Beam* consistently yields higher mean quality than *SMT* for the original subgroup and each alternative. In contrast, the other two values of τ_{abs} do not clearly favor either subgroup-discovery method. For both methods, overfitting, measured by the train-test difference in nWRAcc, is lower for the alternative subgroup descriptions than for the original subgroups. This result may stem from the alternative subgroup descriptions not directly optimizing subgroup quality.

Runtime. Table 4 displays the average runtime for searching original and alternative subgroup descriptions. The search for alternatives is faster for both analyzed subgroup-discovery methods. *Beam* is generally one to two orders of magnitude faster than *SMT*. For *Beam*, runtime tends to decrease with an increasing τ_{abs} and number of alternatives, while *SMT* shows a less clear behavior. In particular, *SMT*'s runtime increases over alternatives for $\tau_{\text{abs}} \in \{1, 2\}$, i.e., settings that allow reusing features from previous subgroup descriptions. In contrast, runtime decreases over alternatives for $\tau_{\text{abs}} = k = 3$, which forbids selecting any feature used before.

6.3 Summary and Discussion

Search methods. The heuristic search methods *Beam* and *BI* were overall best. These two methods from the literature yielded close-to-optimal quality while being significantly faster than our exhaustive, solver-based search method *SMT*. They also beat the quality of the exhaustive competitors *BSD* and *SD-Map*, which required discretizing numeric features beforehand.

Setting a solver timeout allows users to control *SMT*'s runtime but results in suboptimal solutions. *SMT* retains the conceptual advantage that constraints can be added declaratively instead of needing to adapt an algorithmic search procedure to individual constraint types. For example, besides limiting the selected features, one could easily limit the subgroup size, define secondary quality metrics whose values must pass certain thresholds, or add constraints based on domain knowledge. The two particular constraint types studied in this article had the advantage of being antimonotonic, which eased their integration into the three evaluated heuristics.

Heuristic solutions were less prone to overfitting than (training-set-)optimal solutions from exhaustive search. These results highlight the heuristics as serious competitors for any exhaustive search method from the literature, not only our method *SMT*. Finally, we introduced the novel baseline *MORS*, which provided instantaneous, non-trivial lower bounds for subgroup quality.

Feature-cardinality constraints. Imposing feature-cardinality constraints sped up the heuristic search methods as well as *BSD* and *SD-Map*, while the picture for the solver-based search method *SMT* was less clear. Further, this constraint type generally reduced overfitting. Subgroups using as few as $k = 2$ features in their description often yielded already a similar test-set quality as unconstrained subgroups, i.e., using all features. This result speaks for using small feature sets in subgroup descriptions, which may benefit interpretability for users.

Alternative subgroup descriptions. Results for alternative subgroup descriptions strongly depended on two parameters, i.e., the number and dissimilarity of alternatives. With these parameters, users can control alternatives according to their needs. In general, the difference in quality and similarity between the original subgroup and the first alternative was higher than among the first few alternatives. In particular, there may be several promising alternatives from which users may choose one based on further criteria.

Problem definition. As stated in Section 2.1, we focused on datasets with continuous features and a binary target. In general, there are also subgroup-discovery methods specifically for categorical data [34] and other target types [4]. Categorical features can be encoded numerically [57] and thereby integrated into our formalization and experiments. Further, the two constraint types we analyzed are independent of the target type. The chosen binary target is a special case of categorical and numeric targets. Experimentally, switching the target type would require a different subgroup-quality metric and other datasets. In our SMT encoding, one would only need to replace the WRAcc objective (Equation 5) with the new quality metric. Finally, our baseline *MORS* (Section 3.2) would no longer be applicable, though adaptations to other target types may exist.

7 Related Work

In this section, we review related work. We cover subgroup discovery in general and the two analyzed constraint types in particular. Finally, we also discuss adjacent fields.

Subgroup discovery in general. The terms 'subgroup' and 'subgroup discovery' have different meanings in different communities, and similar concepts have been reinvented over time. Our problem definition (Section 2.1) builds on related work in data mining; see [4, 33, 34, 77] for broad surveys of subgroup discovery in this field. There are also recent works independent from this research [3, 45, 68, 74]. These works are only loosely related to ours since their problem definition differs in the optimization objective, type of subgroup description, and employed constraints.

White-box formulations of subgroup discovery. Nearly all existing subgroup-discovery methods, whether they are exhaustive or heuristic, are problem-specific algorithms. To our knowledge, optimizing subgroup discovery with a general-purpose SMT solver is novel. There are only a few white-box formulations for variants of subgroup discovery [14, 20, 30, 42, 53], which tackle different problem definitions than we do. E.g., they have additional constraints and neither consider feature-cardinality constraints nor alternative subgroup descriptions. Further, their experimental evaluations do not compare with existing heuristic subgroup-discovery methods.

Feature-cardinality constraints for subgroup descriptions. Feature cardinality is a common constraint type [59] and a well-known metric for subgroup complexity [34]. However, it is typically integrated into algorithmic rather than solver-based search methods. [51] formulates a quadratic program to select non-redundant features for subgroup descriptions but only as a subroutine within an algorithmic search. Also, their optimization problem employs real-valued feature weights as decision variables rather than defining a discrete feature selection.

While several empirical studies use one fixed feature-cardinality threshold for subgroup discovery, only a few studies systematically analyze the impact of different thresholds [24, 48, 59, 69]. They all employ a narrower experimental design than our study does, e.g., compare fewer subgroup-discovery methods. The broadest of these analyses [59] evaluates three subgroup-discovery methods, including beam and exhaustive search. However, the study focuses on strategies for discretizing numeric data instead of analyzing feature-cardinality constraints in detail. The authors compare $k \in \{1, 2, 3, 4\}$ but not an unconstrained setting. They use 13 datasets, ten of which have at most ten numeric features, while we employ more and higher-dimensional datasets.

Alternative subgroup descriptions. To our knowledge, alternative subgroup descriptions in the sense of our article are novel. Existing approaches for discovering diverse sets of subgroups aim to *minimize* the overlap of contained data objects [4, 11, 15, 54, 76], thereby attempting to cover different regions in the dataset. In contrast, we aim to *maximize* the set similarity of contained data objects to a given subgroup while using different features (Definition 9). Only a few related approaches consider alternatives regarding the subgroup descriptions, which we discuss next.

[76] proposes six strategies to foster diverse subgroup sets. Two of these strategies target diverse subgroup descriptions while optimizing subgroup quality. Both strategies aim at simultaneous rather than sequential search, do not optimize subgroup similarity, and are integrated only into beam search. Finally, they give users less control than our dissimilarity parameter τ does: One strategy excludes subgroup descriptions that have the same quality and differ in only one condition from an existing subgroup description. The other uses a global upper bound on how often a feature may be selected in a subgroup set rather than controlling pairwise dissimilarity.

[52] introduces the notion of *diverse top-k characteristic lists*, which are sets of lists, each containing multiple patterns, e.g., subgroups. The same pattern description must not appear in two lists, but any other overlap of descriptions is allowed. Within lists, patterns should be diverse in terms of data objects covered.

[13] introduces the concept of *equivalent subgroup descriptions of minimal length*, which are stricter than alternative subgroup descriptions. In particular, the former must cover exactly the same set of data objects, like our notion of perfect alternative subgroup descriptions (Definition 10), instead of maximizing similarity. Further, a subset of the original feature set must be found instead of using a different feature set under a dissimilarity constraint. The authors prove $\mathcal{NP}$-hardness, on which our complexity proof for perfect-subgroup discovery (Proposition 4) builds. Additionally, they propose two search algorithms but do not pursue a solver-based search.

Redescription mining aims to find pairs or sets of descriptions that cover exactly or approximately the same data objects [25, 70]. Our notion of alternative subgroup descriptions pursues a similar goal. However, the search for redescriptions is simultaneous and unsupervised [70], while we search alternatives sequentially and start with an original subgroup found supervised, i.e., optimizing subgroup quality. Further, the dissimilarity criteria for redescriptions differ from ours, e.g., having features pre-partitioned into non-overlapping sets [25, 26, 60] or requiring only one arbitrary part of the description to differ [67], while we give users control with the dissimilarity threshold τ. Also, redescriptions may use more complex patterns than subgroup descriptions, e.g., more logical operators than only AND ($\wedge$) to combine conditions over features.

Further related work. There are white-box formulations for various classification models like decision trees, decision sets, and decision lists [36, 64, 75, 79]. These models also use conjunctions of conditions to form descriptive rules, but they try to describe the dataset globally instead of focusing on individual interesting regions like subgroups.

Considering constraints in data mining is a general theme beyond subgroup discovery; see [27] for a survey. Similarly, finding alternative or diverse solutions is a concern in various fields, e.g., clustering [9], subspace search [23], and explainable-AI paradigms like counterfactuals [29], criticisms [40], and semifactuals [1]. Generally, these fields have considerably different problem definitions than subgroup discovery, so we only discuss two examples shortly:

(1) In the field of feature selection, [7, 8] propose a white-box formulation of finding alternative feature sets. However, traditional feature selection differs from subgroup discovery by 'only' selecting the features, which are then used by a prediction model, instead of directly determining bounds on feature values to define a model.

(2) [37, 61, 63, 73] propose approaches for diverse counterfactual explanations. Counterfactuals are data objects that are as similar as possible to a given data object but with a different prediction of a given classifier. In contrast, alternative subgroup descriptions aim at similar predictions for all data objects but a different feature selection.

8 Conclusion

Subgroup-discovery methods constitute an important category of interpretable machine-learning models. This study investigated two constraint types for subgroup discovery: on the number of selected features and for alternative subgroup descriptions. We proved $\mathcal{NP}$-completeness for the corresponding optimization problems. We also studied integrating these constraint types into a novel SMT formulation of subgroup discovery and existing heuristic search methods. Finally, we conducted experiments with eight subgroup-discovery methods and 27 binary-classification datasets. Overall, two heuristic search methods provided the best combination of runtime and subgroup quality, while the performance of exhaustive search methods was unsatisfactory.

Future work could apply our domain-independent methods to specific use cases and interpret the resulting subgroup descriptions qualitatively, i.e., from the domain perspective.

Acknowledgments

We want to thank Klemens Böhm for feedback on an earlier version of this article.

References

[1] André Artelt and Barbara Hammer. 2022. "Even if ..." – Diverse Semifactual Explanations of Reject. In *Proc. SSCI* (Singapore, Singapore). 854–859. doi:10.1109/SSCI51031.2022.10022139

[2] Vadim Arzamasov and Klemens Böhm. 2021. REDS: Rule Extraction for Discovering Scenarios. In *Proc. SIGMOD* (Virtual Event, China). 115–128. doi:10.1145/3448016.3457301

[3] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. 2019. Assessing and Remedying Coverage for a Given Dataset. In *Proc. ICDE* (Macao, China). 554–565. doi:10.1109/ICDE.2019.00056

[4] Martin Atzmueller. 2015. Subgroup discovery. *WIREs Data Min. Knowl. Disc.* 5, 1 (2015), 35–49. doi:10.1002/widm.1144

[5] Martin Atzmueller and Florian Lemmerich. 2009. Fast Subgroup Discovery for Continuous Target Concepts. In *Proc. ISMIS* (Prague, Czech Republic). 35–44. doi:10.1007/978-3-642-04125-9_7

[6] Martin Atzmueller and Frank Puppe. 2006. SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery. In *Proc. PKDD* (Berlin, Germany). 6–17. doi:10.1007/11871637_6

[7] Jakob Bach. 2025. Finding Optimal Diverse Feature Sets with Alternative Feature Selection. arXiv:2307.11607v3 [cs.LG]. doi:10.48550/arXiv.2307.11607

[8] Jakob Bach and Klemens Böhm. 2024. Alternative feature selection with user control. *Int. J. Data Sci. Anal.* (2024). doi:10.1007/s41060-024-00527-8

[9] James Bailey. 2014. Alternative Clustering Analysis: A Review. In *Data Clustering: Algorithms and Applications* (1 ed.). CRC Press, Chapter 21, 535–550. doi:10.1201/9781315373515

[10] Clark Barrett and Cesare Tinelli. 2018. Satisfiability Modulo Theories. In *Handbook of Model Checking* (1 ed.). Springer, Chapter 11, 305–343. doi:10.1007/978-3-319-10575-8_11

[11] Adnene Belfodil, Aimene Belfodil, Anes Bendimerad, Philippe Lamarre, Céline Robardet, Mehdi Kaytoue, and Marc Plantevit. 2019. FSSD - A Fast and Efficient Algorithm for Subgroup Set Discovery. In *Proc. DSAA* (Washington, DC, USA). 91–99. doi:10.1109/DSAA.2019.00023

[12] Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. 2015. νZ - An Optimizing SMT Solver. In *Proc. TACAS* (London, United Kingdom). 194–199. doi:10.1007/978-3-662-46681-0_14

[13] Mario Boley and Henrik Grosskreutz. 2009. Non-redundant Subgroup Discovery Using a Closure System. In *Proc. ECML PKDD* (Bled, Slovenia). 179–194. doi:10.1007/978-3-642-04180-8_29

[14] Tibérius O. Bonates, Peter L. Hammer, and Alexander Kogan. 2008. Maximum patterns in datasets. *Discrete Appl. Math.* 156, 6 (2008), 846–861. doi:10.1016/j.dam.2007.06.004

[15] Guillaume Bosc, Jean-François Boulicaut, Chedy Raïssi, and Mehdi Kaytoue. 2018. Anytime discovery of a diverse set of patterns with Monte Carlo tree search. *Data Min. Knowl. Disc.* 32, 3 (2018), 604–650. doi:10.1007/s10618-017-0547-5

[16] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* 8, 8 (2019). doi:10.3390/electronics8080832

[17] Seung-Seok Choi, Sung-Hyuk Cha, and Charles C. Tappert. 2010. A Survey of Binary Similarity and Distance Measures. *J. Syst. Cybern. Inf.* 8, 1 (2010), 43–48. http://www.iiisci.org/Journal/pdv/sci/pdfs/GS315JG.pdf

[18] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An Efficient SMT Solver. In *Proc. TACAS* (Budapest, Hungary). 337–340. doi:10.1007/978-3-540-78800-3_24

[19] Rodney G. Downey, Michael R. Fellows, and Ulrike Stege. 1997. Parameterized Complexity: A Framework for Systematically Confronting Computational Intractability. In *Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future* (Štiřín Castle, Czech Republic). 49–99. doi:10.1090/dimacs/049/04

[20] Jonathan Eckstein, Peter L. Hammer, Ying Liu, Mikhail Nediak, and Bruno Simeone. 2002. The Maximum Box Problem and its Application to Data Analysis. *Comput. Optim. Appl.* 23, 3 (2002), 285–298. doi:10.1023/A:1020546910706

[21] Stefano Ermon, Carla Gomes, and Bart Selman. 2012. Uniform Solution Sampling Using a Constraint Solver As an Oracle. In *Proc. UAI* (Catalina Island, CA, USA). 255–264. https://www.auai.org/uai2012/papers/160.pdf

[22] Cyril Esnault, May-Line Gadonna, Maxence Queyrel, Alexandre Templier, and Jean-Daniel Zucker. 2020. Q-Finder: An Algorithm for Credible Subgroup Discovery in Clinical Data Analysis – An Application to the International Diabetes Management Practice Study. *Front. Artif. Intell.* 3 (2020). doi:10.3389/frai.2020.559927

[23] Edouard Fouché, Florian Kalinke, and Klemens Böhm. 2021. Efficient subspace search in data streams. *Inf. Syst.* 97 (2021). doi:10.1016/j.is.2020.101705

[24] Jerome H. Friedman and Nicholas I. Fisher. 1999. Bump hunting in high-dimensional data. *Stat. Comput.* 9, 2 (1999), 123–143. doi:10.1023/A:1008894516817

[25] Esther Galbrun and Pauli Miettinen. 2017. *Redescription Mining* (1 ed.). Springer. doi:10.1007/978-3-319-72889-6

[26] Arianna Gallo, Pauli Miettinen, and Heikki Mannila. 2008. Finding Subgroups having Several Descriptions: Algorithms for Redescription Mining. In *Proc. SDM* (Atlanta, GA, USA). 334–345. doi:10.1137/1.9781611972788.30

[27] Valerio Grossi, Andrea Romei, and Franco Turini. 2017. Survey on using constraints in data mining. *Data Min. Knowl. Disc.* 31, 2 (2017), 424–464. doi:10.1007/s10618-016-0480-z

[28] Henrik Grosskreutz and Stefan Rüping. 2009. On subgroup discovery in numerical domains. *Data Min. Knowl. Disc.* 19, 2 (2009), 210–226. doi:10.1007/s10618-009-0136-3

[29] Riccardo Guidotti. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Min. Knowl. Disc.* (2022). doi:10.1007/s10618-022-00831-6

[30] Tias Guns, Siegfried Nijssen, and Luc De Raedt. 2011. Itemset mining: A constraint programming perspective. *Artif. Intell.* 175, 12-13 (2011), 1951–1983. doi:10.1016/j.artint.2011.05.002

[31] Isabelle Guyon and André Elisseeff. 2003. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* 3, Mar (2003), 1157–1182. https://www.jmlr.org/papers/v3/guyon03a.html

[32] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining Frequent Patterns without Candidate Generation. *ACM SIGMOD Rec.* 29, 2 (2000), 1–12. doi:10.1145/335191.335372

[33] Sumyea Helal. 2016. Subgroup Discovery Algorithms: A Survey and Empirical Evaluation. *J. Comput. Sci. Technol.* 31, 3 (2016), 561–576. doi:10.1007/s11390-016-1647-1

[34] Francisco Herrera, Cristóbal José Carmona, Pedro González, and María José Del Jesus. 2011. An overview on subgroup discovery: foundations and applications. *Knowl. Inf. Syst.* 29, 3 (2011), 495–525. doi:10.1007/s10115-010-0356-2

[35] Dan Hudson and Martin Atzmueller. 2023. Subgroup Discovery with SD4Py. In *Proc. ECAI Workshops* (Kraków, Poland). 338–348. doi:10.1007/978-3-031-50396-2_19

[36] Alexey Ignatiev, Joao Marques-Silva, Nina Narodytska, and Peter J. Stuckey. 2021. Reasoning-Based Learning of Interpretable ML Models. In *Proc. IJCAI* (Montreal, Canada). 4458–4465. doi:10.24963/ijcai.2021/608

[37] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-Agnostic Counterfactual Explanations for Consequential Decisions. In *Proc. AISTATS* (Online). 895–905. https://proceedings.mlr.press/v108/karimi20a.html

[38] Richard M. Karp. 1972. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations* (1 ed.). Plenum Press, Chapter 9, 85–103. doi:10.1007/978-1-4684-2001-2_9

[39] Christoph Kiefer, Florian Lemmerich, Benedikt Langenberg, and Axel Mayer. 2022. Subgroup discovery in structural equation models. *Psychol. Methods* (2022). doi:10.1037/met0000524

[40] Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. 2016. Examples are not Enough, Learn to Criticize! Criticism for Interpretability. In *Proc. NIPS* (Barcelona, Spain). https://proceedings.neurips.cc/paper_files/paper/2016/hash/

5680522b8e2bb01943234bce7bf84534-Abstract.html

[41] Mi-Young Kim, Shahin Atakishiyev, Housam Khalifa Bashier Babiker, Nawshad Farruque, Randy Goebel, Osmar R. Zaïane, Mohammad-Hossein Motallebi, Juliano Rabelo, Talat Syed, Hengshuai Yao, and Peter Chun. 2021. A Multi-Component Framework for the Analysis and Design of Explainable Artificial Intelligence. *Mach. Learn. Knowl. Extr.* 3, 4 (2021), 900–921. doi:10.3390/make3040045

[42] Gökberk Koçak, Özgür Akgün, Tias Guns, and Ian Miguel. 2020. Exploiting Incomparability in Solution Dominance: Improving General Purpose Constraint-Based Mining. In *Proc. ECAI* (Santiago de Compostela, Spain). 331–338. doi:10.3233/FAIA200110

[43] Jan Kwakkel. 2017. The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. *Environ. Modell. Software* 96 (2017), 239–250. doi:10.1016/j.envsoft.2017.06.054

[44] Nada Lavrač, Peter Flach, and Blaz Zupan. 1999. Rule Evaluation Measures: A Unifying View. In *Proc. ILP* (Bled, Slovenia). 174–185. doi:10.1007/3-540-48751-4_17

[45] Connor Lawless, Jayant Kalagnanam, Lam M. Nguyen, Dzung Phan, and Chandra Reddy. 2022. Interpretable Clustering via Multi-Polytope Machines. In *Proc. AAAI* (Virtual Event). 7309–7316. doi:10.1609/aaai.v36i7.20693

[46] Florian Lemmerich, Martin Atzmueller, and Frank Puppe. 2016. Fast exhaustive subgroup discovery with numerical target concepts. *Data Min. Knowl. Disc.* 30, 3 (2016), 711–762. doi:10.1007/s10618-015-0436-8

[47] Florian Lemmerich and Martin Becker. 2019. pysubgroup: Easy-to-Use Subgroup Discovery in Python. In *Proc. ECML PKDD* (Dublin, Ireland). 658–662. doi:10.1007/978-3-030-10997-4_46

[48] Florian Lemmerich, Mathias Rohlfs, and Martin Atzmueller. 2010. Fast Discovery of Relevant Subgroup Patterns. In *Proc. FLAIRS* (Daytona Beach, FL, USA). 428–433. https://aaai.org/papers/flairs-2010-1262/

[49] Haobo Li, Yunxia Liu, Ke Chen, Johannes T. Margraf, Youyong Li, and Karsten Reuter. 2021. Subgroup Discovery Points to the Prominent Role of Charge Transfer in Breaking Nitrogen Scaling Relations at Single-Atom Catalysts on VS2. *ACS Catal.* 11, 13 (2021), 7906–7914. doi:10.1021/acscatal.1c01324

[50] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2017. Feature Selection: A Data Perspective. *ACM Comput. Surv.* 50, 6 (2017). doi:10.1145/3136625

[51] Rui Li, Robert Perneczky, Alexander Drzezga, and Stefan Kramer. 2015. Efficient redundancy reduced subgroup discovery via quadratic programming. *J. Intell. Inf. Syst.* 44, 2 (2015), 271–288. doi:10.1007/s10844-013-0284-1

[52] Antonio Lopez-Martinez-Carrasco, Hugo M. Proença, Jose M. Juarez, Matthijs van Leeuwen, and Manuel Campos. 2023. Discovering Diverse Top-K Characteristic Lists. In *Proc. IDA* (Louvain-la-Neuve, Belgium). 262–273. doi:10.1007/978-3-031-30047-9_21

[53] Quentin Louveaux and Sébastien Mathieu. 2014. A combinatorial branch-and-bound algorithm for box search. *Discrete Optim.* 13 (2014), 36–48. doi:10.1016/j.disopt.2014.05.001

[54] Tarcísio Lucas, Renato Vimieiro, and Teresa Ludermir. 2018. SSDP+: A Diverse and More Informative Subgroup Discovery Approach for High Dimensional Data. In *Proc. CEC* (Rio de Janeiro, Brazil). doi:10.1109/CEC.2018.8477855

[55] Michael Mampaey, Siegfried Nijssen, Ad Feelders, and Arno Knobbe. 2012. Efficient Algorithms for Finding Richer Subgroup Descriptions in Numeric and Nominal Data. In *Proc. ICDM* (Brussels, Belgium). 499–508. doi:10.1109/ICDM.2012.117

[56] Romain Mathonat, Diana Nurbakova, Jean-François Boulicaut, and Mehdi Kaytoue. 2021. Anytime Subgroup Discovery in High Dimensional Numerical Data. In *Proc. DSAA* (Porto, Portugal). doi:10.1109/DSAA53316.2021.9564223

[57] Federico Matteucci, Vadim Arzamasov, and Klemens Böhm. 2023. A benchmark of categorical encoders for binary classification. In *Proc. NeurIPS* (New Orleans, LA, USA). 54855–54875. https://proceedings.neurips.cc/paper_files/paper/2023/hash/ac01e21bb14609416760f790dd8966ae-Abstract-Datasets_and_Benchmarks.html

[58] Marvin Meeng, Wouter Duivesteijn, and Arno Knobbe. 2014. ROCsearch – An ROC-guided Search Strategy for Subgroup Discovery. In *Proc. SDM* (Philadelphia, PA, USA). 704–712. doi:10.1137/1.9781611973440.81

[59] Marvin Meeng and Arno Knobbe. 2021. For real: a thorough look at numeric attributes in subgroup discovery. *Data Min. Knowl. Disc.* 35, 1 (2021), 158–212. doi:10.1007/s10618-020-00703-x

[60] Matej Mihelčić and Adrian Satja Kurdija. 2023. On the complexity of redescription mining. *Theor. Comput. Sci.* 944 (2023). doi:10.1016/j.tcs.2022.12.023

[61] Kiarash Mohammadi, Amir-Hossein Karimi, Gilles Barthe, and Isabel Valera. 2021. Scaling Guarantees for Nearest Counterfactual Explanations. In *Proc. AIES* (Virtual Event, USA). 177–187. doi:10.1145/3461702.3462514

[62] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. 2020. Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges. In *Proc. ECML PKDD Workshops* (Ghent, Belgium). 417–431. doi:10.1007/978-3-030-65965-3_28

[63] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *Proc. FAT** (Barcelona, Spain). 607–617. doi:10.1145/3351095.3372850

[64] Nina Narodytska, Alexey Ignatiev, Filipe Pereira, and Joao Marques-Silva. 2018. Learning Optimal Decision Trees with SAT. In *Proc. IJCAI* (Stockholm, Sweden). 1362–1368. doi:10.24963/ijcai.2018/189

[65] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang. 1998. Exploratory Mining and Pruning Optimizations of Constrained Associations Rules. *ACM SIGMOD Rec.* 27, 2 (1998), 13–24. doi:10.1145/276305.276307

[66] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. 2017. PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Min.* 10 (2017). doi:10.1186/s13040-017-0154-4

[67] Laxmi Parida and Naren Ramakrishnan. 2005. Redescription Mining: Structure Theory and Algorithms. In *Proc. AAAI* (Pittsburgh, PA, USA). 837–844. https://aaai.org/papers/00837-aaai05-132-redescription-mining-structure-theory-and-algorithms/

[68] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence. In *Proc. SIGMOD* (Virtual Event, China). 1400–1412. doi:10.1145/3448016.3457284

[69] Hugo M. Proença, Peter Grünwald, Thomas Bäck, and Matthijs van Leeuwen. 2022. Robust subgroup discovery: Discovering subgroup lists using *MDL*. *Data Min. Knowl. Disc.* 36, 5 (2022), 1885–1970. doi:10.1007/s10618-022-00856-x

[70] Naren Ramakrishnan, Deept Kumar, Bud Mishra, Malcolm Potts, and Richard F. Helm. 2004. Turning CARTwheels: An Alternating Algorithm for Mining Redescriptions. In *Proc. KDD* (Seattle, WA, USA). 266–275. doi:10.1145/1014052.1014083

[71] Youcef Remil, Anes Bendimerad, Romain Mathonat, Philippe Chaleat, and Mehdi Kaytoue. 2021. "What makes my queries slow?": Subgroup Discovery for SQL Workload Analysis. In *Proc. ASE* (Melbourne, Australia). 642–652. doi:10.1109/ASE51524.2021.9678915

[72] Joseph D. Romano, Trang T. Le, William La Cava, John T. Gregg, Daniel J. Goldberg, Natasha L. Ray, Praneel Chakraborty, Daniel Himmelstein, Weixuan Fu, and Jason H. Moore. 2021. PMLB v1.0: An open source dataset collection for benchmarking machine learning methods. arXiv:2012.00058v3 [cs.LG]. doi:10.48550/arXiv.2012.00058

[73] Chris Russell. 2019. Efficient Search for Diverse Coherent Explanations. In *Proc. FAT** (Atlanta, GA, USA). 20–28. doi:10.1145/3287560.3287569

[74] Svetlana Sagadeeva and Matthias Boehm. 2021. SliceLine: Fast, Linear-Algebra-based Slice Finding for ML Model Debugging. In *Proc. SIGMOD* (Virtual Event, China). 2290–2299. doi:10.1145/3448016.3457323

[75] Pouya Shati, Eldan Cohen, and Sheila McIlraith. 2021. SAT-Based Approach for Learning Optimal Decision Trees with Non-Binary Features. In *Proc. CP* (Montpellier, France (Virtual Conference)). doi:10.4230/LIPIcs.CP.2021.50

[76] Matthijs van Leeuwen and Arno Knobbe. 2012. Diverse subgroup set discovery. *Data Min. Knowl. Disc.* 25, 2 (2012), 208–242. doi:10.1007/s10618-012-0273-y

[77] Sebastián Ventura and José María Luna. 2018. *Subgroup Discovery* (1 ed.). Springer, Chapter 4, 71–98. doi:10.1007/978-3-319-98140-6_4

[78] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y. Lim. 2019. Designing Theory-Driven User-Centric Explainable AI. In *Proc. CHI* (Glasgow, United Kingdom). doi:10.1145/3290605.3300831

[79] Jinqiang Yu, Alexey Ignatiev, Peter J. Stuckey, and Pierre Le Bodic. 2021. Learning Optimal Decision Sets and Lists with SAT. *J. Artif. Intell. Res.* 72 (2021), 1251–1279. doi:10.1613/jair.1.12719

[80] Cristina Zuheros, Eugenio Martínez-Cámara, Enrique Herrera-Viedma, Iyad A. Katib, and Francisco Herrera. 2023. Explainable Crowd Decision Making methodology guided by expert natural language opinions based on Sentiment Analysis with Attention-based Deep Learning and Subgroup Discovery. *Inf. Fusion* 97 (2023). doi:10.1016/j.inffus.2023.101821

Received October 2024; revised January 2025; accepted February 2025