# Non-Redundant Subgroup Discovery Using a Closure System

Mario Boley and Henrik Grosskreutz

Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, Germany
{mario.boley,henrik.grosskreutz}@iais.fraunhofer.de

**Abstract.** Subgroup discovery is a local pattern discovery task, in which descriptions of subpopulations of a database are evaluated against some quality function. As standard quality functions are functions of the described subpopulation, we propose to search for equivalence classes of descriptions with respect to their extension in the database rather than individual descriptions. These equivalence classes have unique maximal representatives forming a closure system. We show that minimum cardinality representatives of each equivalence class can be found during the enumeration process of that closure system without additional cost, while finding a minimum representative of a single equivalence class is **NP**-hard. With several real-world datasets we demonstrate that search space and output are significantly reduced by considering equivalence classes instead of individual descriptions and that the minimum representatives constitute a family of subgroup descriptions that is of same or better expressive power than those generated by traditional methods.

## 1 Introduction

Subgroup discovery [2, 12, 17] is a local pattern discovery task: descriptions of subpopulations of a database are evaluated against some real-valued quality function, and those descriptions exceeding some given minimum quality are returned to the user. The quality functions commonly used in this course like Piatetsky-Shapiro, binomial test, or Gini-index (see [12] for a list) are *functions of the extension* of a subgroup description. Traditional subgroup discovery algorithms, however, search in the space of subgroup descriptions, usually conjunctions of attribute/value equality constraints, rather than in the space of extensions. Since many descriptions can have an identical extension on the given data, this may lead to (i) many redundant evaluations of the quality function and (ii) to a result set that contains multiple descriptions of the same subpopulation.

In contrast we propose to consider extension based *equivalence classes* of subgroup descriptions rather than individual descriptions. Thereby our algorithms implicitly search in the space of subgroup extensions, and, consequently, they have a potentially reduced search space and return at most one description of each extension, *a representative*, to the user. For this purpose we propose to use descriptions with a *minimum number of constraints*, i.e., a minimum representative. This choice is motivated by the common hypothesis that (i) short

descriptions are easier to interpret than long descriptions and (ii) in scenarios in which subgroups are used as building blocks for global models short descriptions lead to better generalization. In summary we consider the following computational problem:

*Problem 1 (*NON-REDUNDANT-SUBGROUP-DISCOVERY*). Given* a dataset, a quality function $q$, and a minimum quality threshold $q^*$, *list* a result set $\mathcal{R}$ of subgroup descriptions that satisfies

**completeness,** i.e., for all subgroup descriptions $H$ with $q(H) \geq q^*$ there is an $H' \in \mathcal{R}$ that has the same extension as $H$,

**non-redundancy,** i.e., for all pairs of distinct subgroup descriptions $H, H' \in \mathcal{R}$ the extensions of $H$ and $H'$ are distinct, and

**representative minimality,** i.e., for all listed descriptions $H \in \mathcal{R}$ there is no shorter description $H'$ having the same extension.

In addition we also discuss the standard problem variations of mining only representatives of the *top-k* quality equivalence classes as well as mining only classes having a representative not exceeding a given *length-limit*.

*Results and Contribution* We formalize extension based equivalence classes and show that they theoretically can subsume an exponential number of individual subgroup descriptions. Thus, searching equivalence classes rather than individual descriptions has the potential to reduce search space and output tremendously. Indeed, as we show in an empirical study, a significant reduction can also be observed on ten well-known real-world datasets.

While each equivalence class has a unique maximal representative, we show that the number of minimal and minimum representatives can grow exponentially in the length of that maximal description, and it is **NP**-hard, given some description, to compute an equivalent description of minimum length. This result is complemented by our observation that a simple greedy strategy approximates a minimum representative within a logarithmic factor.

We use this approximation technique together with the fact that the unique maximal representatives are forming a closure system to develop a first algorithmic solution to Problem 1 that builds on any of the known algorithms that can enumerate closure systems combined with an anti-monotone pruning condition. As an alternative we present an algorithm that directly traverses the equivalence classes via the exact minimum representatives. This approach bypasses the hardness of computing minimum representatives by building them inductively from one another. It comes, however, at the cost of additional memory requirements.

In a concluding empirical evaluation we compare both approaches to each other and to traditional exhaustive subgroup discovery. We assess good performance, as well as a surprisingly good predictive power given that the method has not been optimized towards this goal.

*Prior Work* Addressing output redundancy is a concern of subgroup discovery research right from the start (see, e.g., [8]). More recent approaches include successive weighted covering [13] and the removal of irrelevant descriptions [7]. For the first method it is important to note that it can be combined with our's rather than being an excluding alternative. The second approach is closely related to this work in that it also uses the closure system of maximal representatives. While we are interested in listing minimum representatives of each equivalence class, their focus is on discarding irrelevant classes.

Traditional subgroup miners usually exploit an optimistic estimator of their quality function in order to make the approach of searching in the space of descriptions feasible. Hence they create an anti-monotone search space that contains the family of all interesting descriptions. While our algorithms use equivalence classes instead of individual descriptions, all the optimistic estimator techniques including recent findings [10] can still be applied. Consequently, our algorithms always use a condensed version of traditional method's search spaces.

Methodologically our work is directly related to formal concept analysis (e.g., [6]) and closed set mining [3, 14, 16] because the maximal representatives together with their equivalence classes form a concept lattice. While our algorithms build on closure system enumeration techniques, we are only interested in minimum representatives. This is in contrast to closed set mining where algorithms seek to list all maximal (closed) members or minimal members (generators or free sets).

## 2 Basic Definitions

Throughout this work we denote "elementary objects" by non-capital letters, e.g., $e$, sets of elementary objects by capital letters, e.g, $E$ and families, i.e., sets, of sets by calligraphic letters, e.g., $\mathcal{E}$. In particular the power set of some set $E$ is denoted by $\mathcal{P}(E)$. The symbol "$\subset$" denotes the strict subset relation between sets. For a family $\mathcal{S}$ the terms **minimal** and **maximal** refer to the subset relation, i.e., a set $S \in \mathcal{S}$ is a minimal element of $\mathcal{S}$ if there no strict subset $S' \subset S$ that is also an element of $\mathcal{S}$. In contrast, the term **minimum** is used with respect to the cardinality of a set, i.e., $S \in \mathcal{S}$ is called minimum element of $\mathcal{S}$ if there is no $S' \in \mathcal{S}$ with $|S'| < |S|$.

*Subgroup Descriptions* Let $\mathcal{A} = A_1, \ldots, A_n$ be a sequence of $n$ sets we refer to as **attributes**. A **data record** over $\mathcal{A}$ is an $n$-tuple $D = (a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n$, for which we denote its $i$-th component by $D(i) = a_i$. A **dataset** $\mathcal{D}$ over $\mathcal{A}$ is a multiset of data records over $\mathcal{A}$. Note that we do not consider labeled data respectively target attributes at this points. Labels are introduced in the paragraph about subgroup quality below.

The subgroup description language considered in this work is the language of conjunctions of attribute/value equality constraints. We formalize this as follows: a **constraint** over $\mathcal{A}$ is an expression $(A_i = v)$ with $i \in \{1, \ldots, n\}$ and $v \in A_i$. The set of all such constraints is denoted $\boldsymbol{C}_{\mathcal{A}}$. The family of **subgroup descriptions** over $\mathcal{A}$, i.e., the language of conjunctions of such constraints, is

then $\mathcal{L}_{\mathcal{A}} = \mathcal{P}(\boldsymbol{C}_{\mathcal{A}})$. In the following we drop the index $\mathcal{A}$ whenever it is clear from the context.

The semantic of conjunctions arises through the following interpretation of subgroup descriptions: let $\mathcal{D}$ be a dataset over $\mathcal{A}$. A datarecord $D \in \mathcal{D}$ is said to **support** a subgroup description $H \in \mathcal{L}$, denoted $D \models H$, if for all $(A_i = v) \in H$ it holds that $D(i) = v$. Then the **extension** of $H$ in $\mathcal{D}$, denoted by $\mathcal{D}[H]$, is the submultiset of $\mathcal{D}$ containing the data records that support $H$. Extensions are anti-monotone with respect to the subset relation, i.e., it holds that $H \subseteq H' \Rightarrow \mathcal{D}[H] \supseteq \mathcal{D}[H']$.

*Subgroup Quality* For the purpose of this work we simply regard a **quality function** as a map $q\colon \mathcal{L} \to \mathbb{R}$ and an **optimistic estimator** for $q$ as a map $\hat{q}\colon \mathcal{L} \to \mathbb{R}$ satisfying for all $H \subseteq H' \in \mathcal{L}$ that $\hat{q}(H) \geq q(H')$. Given a **quality threshold** $q^*$ the family of **interesting subgroup descriptions** is $\mathcal{R} = \{H \in \mathcal{L}\colon\ q(H) \geq q^*\}$. Usually, of course, a quality function depends on the given data. For instance for a **binary labeled dataset** $\mathcal{D}$, i.e., a dataset with associated labels $l(D) \in \{+, -\}$ for all $D \in \mathcal{D}$ a commonly used quality function (and the one used in our experiments) is the **binomial test quality function**

$$q(H) = \sqrt{\frac{|\mathcal{D}[H]|}{|\mathcal{D}|}} \left( \frac{|\mathcal{D}^+[H]|}{|\mathcal{D}[H]|} - \frac{|\mathcal{D}^+|}{|\mathcal{D}|} \right) \ ,$$

where $\mathcal{D}^+ = \{D \in \mathcal{D}\colon\ l(D) = +\}$ denotes the dataset of all $+$-labeled data records. As optimistic estimator for the binomial test quality function we used $\hat{q}(H) = \sqrt{|\mathcal{D}[H]| \,/\, |\mathcal{D}|}\,(1 - |\mathcal{D}^+| \,/\, |\mathcal{D}|)$. In the following, however, the concrete form of $q$ and $\hat{q}$ is not considered. We regard them as given blackboxes that "encapsulate" the data and rely only on the following two requirements:

1. For a given **minimum quality threshold** $q^* \in \mathbb{R}$ the **search space** $\mathcal{S} = \{H \in \mathcal{L}\colon\ \hat{q}(H) \geq q^*\}$ defined by $\hat{q}$ is **anti-monotone**, i.e., for all $H \subseteq H' \subseteq \boldsymbol{C}$ it holds that $H' \in \mathcal{S}$ implies $H \in \mathcal{S}$.
2. The maps $q(H)$ and $\hat{q}(H)$ both are *functions of the extension* of $H$ in the dataset, i.e., $\mathcal{D}[H] = \mathcal{D}[H']$ implies $q(H) = q(H')$.

The first requirement follows from the definition of optimistic estimators, and the second is true for the usually employed quality functions and their estimators.

## 3 Extension Equivalence and Compression

In this section we formally introduce equivalence classes of subgroup descriptions and investigate their potential in reducing search space and output of subgroup discovery. Unless explicitly mentioned otherwise, for the remainder of this article we assume that $\mathcal{D}$ is a dataset of size $m$ over $n$ attributes $\mathcal{A}$. The central notion of equivalence is:

**Definition 1 (Description Equivalence).** *Two subgroup descriptions $H, H' \in \mathcal{L}$ are **equivalent** (with respect to the dataset $\mathcal{D}$), denoted by $H \equiv H'$, if they have an identical extension on $\mathcal{D}$, i.e., $\mathcal{D}[H] = \mathcal{D}[H']$.*

Clearly, $\equiv$ is an equivalence relation on $\mathcal{L}$. For a $H \in \mathcal{L}$ we denote its **equivalence class** with respect to $\equiv$, i.e., $\{H' \in \mathcal{L} \colon H \equiv H'\}$, by $[H]$. For a family of subgroup descriptions $\mathcal{H} \subseteq \mathcal{L}$ we denote by $\mathcal{H}^{\equiv}$ the equivalence classes it contains, i.e., $\mathcal{H}^{\equiv} = \{[H] \colon H \in \mathcal{H}\}$.

In order to investigate the potential reduction of search space and output, we now give a general dataset construction that intuitively reflects "worst-case siutations" for traditional subgroup discovery. It leads to several theoretical observations.

**Construction 1** *For even positive integers $n \in \mathbb{N}$ we define the dataset $\mathcal{D}_n$ over $n$ binary attributes $\mathcal{A} = (A_1, \ldots, A_n)$ with $A_i = \{0, 1\}$ for $i \in \{1, \ldots, n\}$ by $\mathcal{D}_n = (D_1, \ldots, D_{n/2+1})$ with $D_{n/2+1} = (1, \ldots, 1)$ and*

$$D_i(j) = \begin{cases} 0, & \text{if } j \in \{2i-1, 2i\} \\ 1, & \text{otherwise} \end{cases} ,$$

*for $i = 1, \ldots, n/2$.*

Table 1 illustrates this construction for $n = 6$ annotated with binary labels. For

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | l |
|---|---|---|---|---|---|---|---|
| $D_1$ | 0 | 0 | 1 | 1 | 1 | 1 | − |
| $D_2$ | 1 | 1 | 0 | 0 | 1 | 1 | − |
| $D_3$ | 1 | 1 | 1 | 1 | 0 | 0 | − |
| $D_4$ | 1 | 1 | 1 | 1 | 1 | 1 | + |

**Table 1.** Construction 1 with $n = 6$

$q^* = 3/8$ only one equivalence class is interesting with respect to the binomial test quality function, i.e, the one containing the descriptions $H$ with $\mathcal{D}[H] = \{D_4\}$. Thus, one solution to Problem 1 for this data is $\{(A_1 = 1), (A_3 = 1), (A_5 = 1)\}$. In contrast there are $3^3 = 27$ alternative descriptions of this extension: for each of the pairs $\{A_1, A_2\}$, $\{A_3, A_4\}$, and $\{A_5, A_6\}$ choose one or both attributes to be constraint to 1. Generally the datasets $\mathcal{D}_n$ witness that the compression rate achieved by considering equivalence classes instead of individual description can grow exponentially in the number of attributes (and data records).

**Theorem 1** *For all positive integers $n \in \mathbb{N}$ there is a dataset $\mathcal{D}$ of size $n/2 + 1$ over $n$ attributes $\mathcal{A}$ and a quality threshold $q^*$ such that the compression rates $|\mathcal{R}| \,/\, |\mathcal{R}^{\equiv}|$ and $|\mathcal{S}| \,/\, |\mathcal{S}^{\equiv}|$ are in $O(\exp(n/2))$.*

In order to investigate the extend of compression that can be achieved in practice, we conducted experiments on ten real-world datasets, which are introduced in more detail in Section 7. Table 2 shows sizes of result families $\mathcal{R}$, compressed results families $\mathcal{R}^{\equiv}$, search spaces $\mathcal{S}$, and compressed search spaces $\mathcal{S}^{\equiv}$ for different quality thresholds $q^*$. The threshold $t100$ varies among the datasets: it

is equal to the quality of the 100th highest quality subgroup description (note that because of ties in the quality $|\mathcal{R}_{t100}|$ can still be greater than 100). The threshold $\epsilon$ is equal to the smallest positive number distinguishable from zero in double precision. The results for $q^* = t100$ give a differentiated impression: rang-

| | credi. | lung-. | lymph | mush. | nurse. | sick | soybe. | splice | tic-t. | vote |
|---|---|---|---|---|---|---|---|---|---|---|
| $\lvert\mathcal{R}_{t100}\rvert$ | 100 | 38K | 124 | 168 | 100 | 128 | 749K | 100 | 113 | 101 |
| $\lvert\mathcal{R}^{\equiv}_{t100}\rvert$ | 83 | 1 | 17 | 12 | 100 | 1 | 1 | 99 | 113 | 101 |
| $\lvert\mathcal{S}_{t100}\rvert$ | 148K | 456M | 12K | 3458 | 103K | >100M | >100M | 398K | 6067 | 3505 |
| $\lvert\mathcal{S}^{\equiv}_{t100}\rvert$ | 87K | 159K | 4176 | 890 | 69K | 8M | 1M | 395K | 5824 | 3465 |
| $\lvert\mathcal{R}_{\epsilon}\rvert$ | 6119K | >100M | 1078K | >100M | 11K | >100M | >100M | >100M | 65K | 3610K |
| $\lvert\mathcal{R}^{\equiv}_{\epsilon}\rvert$ | 175K | 103K | 19K | 105K | 11K | 2M | 2M | >100M | 23K | 82K |
| $\lvert\mathcal{S}_{\epsilon}\rvert$ | 17M | >100M | 26M | >100M | 192K | >100M | >100M | >100M | 129K | 11M |
| $\lvert\mathcal{S}^{\equiv}_{\epsilon}\rvert$ | 385K | 183K | 45K | 228K | 115K | 9M | 3M | >100M | 43K | 227K |

**Table 2.** Uncompressed and compressed result families and search spaces

ing from tremendious compression rates of 749.000 (soybean) to no compression (vote). For decreasing thresholds, however, a significant compression arises for all datasets: while (with one exception) it is tractable to search through all equivalence classes with a potentially positive quality ($q^* = \epsilon$), this is infeasible on most datasets for exhaustive enumeration due to the large number of equivalent descriptions.

## 4 Border Elements

Border elements, i.e., maximal and minimal members of an equivalence class $[H]$, play a special role. They contain all information necessary to check whether some given description is a member of $[H]$. Among the minimal members one can find minimum representatives, one of which we desire as representative for its class. In this section we state some basic but important mathematical and computational properties of the border elements.

The first observation is that every equivalence class has a unique maximal (most specific) element. It is given by the map $\sigma$ introduced in the lemma below.

**Lemma 2 (Pasquier et al. [14])** *For all subgroup descriptions $H \in \mathcal{L}$ it holds that $\sigma(H)$ given by*

$$\sigma(H) = \{(A_i{=}v) \colon 1 \leq i \leq n, \forall D \in \mathcal{D}[H]\,,\, D(i) = v\} \ .$$

*is the unique maximal element of $[H]$, i.e., (i) $H \equiv \sigma(H)$, (ii) for all $H' \in \mathcal{L}$ with $H' \supset \sigma(H)$ it holds that $H \not\equiv H'$, and (iii) $\sigma(H)$ is unique with (i) and (ii).*

While each equivalence class $[H]$ has a unique maximal element $\sigma(H)$, there can be more than one minimal (most general) element of $[H]$. In fact the number of minimal and even that of the minimum representatives can be exponential

in the cardinality of $\sigma(H)$. Again, the datasets $\mathcal{D}_n$ from Construction 1 witness this statement: for all $I \subseteq \{1, \ldots, n/2\}$ the description $H_I$ defined by

$$H_I = \{(A_{2i} = 1) \colon i \in I\} \cup \{(A_{2i-1} = 1)) \colon \{1, \ldots, n\} \setminus I\}$$

is a minimum description of the extension $\{D_{n+1}\}$, and there are $2^{n/2}$ such descriptions. We can conclude:

**Theorem 3** *For all positive integers $n \in \mathbb{N}$ there is a dataset $\mathcal{D}$ of size $n/2+1$ over $n$ attributes $\mathcal{A}$ such that there is an equivalence class with $O(\exp(n/2))$ minimum representatives.*

For Problem 1 we are only interested in constructing one minimum representative per interesting equivalence class. As an isolated task, however, this is intractable. This again contrasts the maximal representatives, which can be computed in time $O(nm)$. In particular the **NP**-hard MIN-SET-COVER problem—*given* a family of subsets $\mathcal{F} \subseteq \mathcal{P}(E)$ with $\bigcup \mathcal{F} = E$, *compute* a minimum subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $\bigcup \mathcal{F}' = E$—polynomially reduces to finding a minimum description. In addition the reduction preserves solution sizes. Thus, even the inapproximability result from [5] carries over to our problem.

**Theorem 4** Given *a subgroup description $H \in \mathcal{L}$, it is*

*(a)* **NP***-hard to* compute *an equivalent subgroup description $G \in [H]$ of minimum length, i.e., $|G| = \min\{|H'| \colon H' \in [H]\}$ and*
*(b) hard*[1] *to compute an approximation $G' \in [H]$ in polynomial time that satisfies $|G'| \leq |G|\,(1-\epsilon) \ln m$ for all $\epsilon > 0$ where $m = |\mathcal{D} \setminus \mathcal{D}[H]|$ is the number of data records not supporting $H$.*

*Proof.* We prove both statements by giving a polynomial time transformation of MIN-SET-COVER instances $\mathcal{F} \subseteq \mathcal{P}(E)$ to a dataset $\mathcal{D}$ over attributes $\mathcal{A}$ and a subgroup description $H \in \mathcal{L}_{\mathcal{A}}$ such that (i) extension equivalent descriptions $H' \in [H]$ correspond to set covers $\mathcal{F}_{H'} \subseteq \mathcal{F}$ of $E$ of same size, i.e., $|H'| = |\mathcal{F}_{H'}|$ and (ii) $|\mathcal{D} \setminus \mathcal{D}[H]|$ is equal to the size of the set cover ground set $|E|$. Part (a) then follows from the **NP**-hardness of MIN-SET-COVER and (b) from the result of [5]. Let $E = \{1, \ldots, m\}$ and $\mathcal{F} = \{S_1, \ldots, S_n\}$ be a set cover instance. Set $\mathcal{A} = \{A_1, \ldots, A_n\}$ with $A_i = \{0, 1\}$ and $\mathcal{D} = D_1, \ldots, D_{m+1}$ with

$$D_i(j) = \begin{cases} 1, & \text{if } i \notin S_j \\ 0, & \text{otherwise} \end{cases}$$

for $i = \{1, \ldots, m\}$ and $D_{m+1} = (1, \ldots, 1)$. Furthermore, choose $H \in \mathcal{L}$ as $H = \{(A_i = 1) \colon 1 \leq i \leq n\}$. Let $H' \in [H]$ be a subgroup description equivalent to $H$. Then it follows from the definitions that $H' \subseteq \sigma(H)$ and $\mathcal{D}'[H'] = \{\}$

[1] Here, hardness means: "as hard as computing a solution to an **NP**-hard problem in time $n^{O(\log \log n)}$ for instances of size $n$."

where $\mathcal{D}' = \mathcal{D} \setminus \mathcal{D}[H]$ denote the datarecords that are not supporting $H$. It follows for $\mathcal{F}_{H'} \subseteq \mathcal{F}$ defined by $\mathcal{F}_{H'} = \{S_i \colon (A_i{=}1) \in H'\}$ that

$$\begin{aligned} &\forall i \in E,\, \exists j,\, i \in S_j \wedge S_j \in \mathcal{F}_{H'} \\ \Leftrightarrow &\forall i \in E,\, \exists (A_j{=}1) \in H',\, D_i(j) = 0 \\ \Leftrightarrow &\forall i \in E,\, D_i \notin \mathcal{D}'[H'] \quad . \end{aligned}$$

That is $H' \equiv H$ if and only if $\mathcal{F}_{H'}$ is a cover of $E$. Moreover, $|H'| = |\mathcal{F}_{H'}|$ as required. □

## 5 Closure System Traversal and Greedy Approximation

Our first algorithmic approach towards solving Problem 1 is motivated by the observation that the transformation used within the proof of Theorem 4 can be reversed. Hence, finding a minimum equivalent representative of a given description $H$ is in fact equivalent to MIN-SET-COVER. Incoporating that inverse transformation into the well-known greedy algorithm for MIN-SET-COVER yields the procedure:

1. **set** $\bar{\mathcal{D}} \leftarrow \mathcal{D} \setminus \mathcal{D}[H]$, $G \leftarrow \emptyset$
2. **while** $\bar{\mathcal{D}}[G] \neq \emptyset$ **set** $G \leftarrow G \cup \{\arg\min_{c \in \sigma(H)} \left|\bar{\mathcal{D}}[G \cup \{c\}]\right|\}$
3. **return** $G$

Slavík [15] found that the **greedy approximation factor** for MIN-SET-COVER is $g(m) = \ln m - \ln\ln m + 0.78$ for instances with a ground set $E$ of size $m$. Taking into account our transformation we have the following result.

**Lemma 5** *Given a subgroup description $H$, a minimum representative of $[H]$ can be approximated in time $O(|\sigma(H)|\ m)$ within $g(m)$ where $m = |\mathcal{D} \setminus \mathcal{D}[H]|$ is the number of data records not supporting $H$.*

Thus, any algorithm that traverses the search space of equivalence classes can be combined with the greedy algorithm to approximately solve Problem 1. Several known algorithms are identified as applicable for that task by another observation: the maximal representatives form a closure system.

**Lemma 6 (Pasquier et al. [14])** *The map $\sigma$ is a closure operator, i.e., it satisfies for all $H, H' \in \mathcal{L}$ that $H \subseteq \sigma(H)$ (extensivity), $H \subseteq H' \Rightarrow \sigma(H) \subseteq \sigma(H')$ (monotonicity), and $\sigma(H) = \sigma(\sigma(H))$ (idempotence).*

There are several efficient algorithms listing all closed sets of a closure operator like the divide and conquer algorithm from formal concept analysis [9]. Adapting a closed frequent itemset miner like LCM [16] to our task is even more natural: we plug in optimistic estimate pruning instead of frequency pruning, and instead of single items we have single constraints. Since there are at most $nm$ valid constraints for a dataset of size $m$ over $n$ attributes, together with the performance of LCM we get the result:

**Theorem 7** *Problem 1 can be solved in time* $O(|\mathcal{S}^{\equiv}|\, n^2m^2)$ *and space* $O(nm)$ *if the representative minimality condition is relaxed to: for all listed descriptions* $H \in \mathcal{R}$ *there is no description* $H'$ *having the same extension with* $|H| > g(|\mathcal{D} \setminus \mathcal{D}[H]|)\,|H'|$.

Note that in case of a constant number of attribute values the bound on the number of constraints boils down to $O(n)$, and consequently the time complexity in the theorem is improved by a factor $m$. The theoretical approximation guarantee of the greedy algorithm, although optimal with regard to Theorem 4, may appear somewhat weak. In practice, however, the worst-case bound is virtually never attained, and the greedy result is usually close to optimum (see Section 7). Moreover, note that the potentially expensive greedy algorithm has to be called only for the returned result equivalence classes and not during the actual traversal of the closure system.

In order to adress the *top-k* problem variant, i.e., to list only representatives of $k$ highest quality classes, only minor changes are necessary: instead of directly printing the interesting subgroup descriptions, collect them in a priority queue with capacity $k$. In this scenario the search space can be reduced significantly by adjusting the $q^*$ threshold whenever a new subgroup description is added to the result queue (and the queue is full).

On the other hand, there is no easy way to include a *length-limit* for additional pruning, i.e., when we are only interested in reprentatives containing no more than $l$ literals, this cannot be exploited for reducing the search space. The reason is that, even if exact minimum reprentatives would be on hand, in general LCM (or any other common closed set miner) does not list the closed sets in ascending order with respect to their minimum equivalent descriptions.

## 6 Inductive Minimum Representative Construction

In this section we present an alternative algorithmic approach for non-redundant subgroup discovery that, intuitively, is based on a breadth-first traversal of the directed graph containing as vertices all equivalence classes that lie in the search space and edges between any two classes $[H] \neq [H']$ such that there is a constraint $c \in \boldsymbol{C}$ with $(H \cup \{c\}) \in [H']$ (note that the existence of such a constraint is independent of the chosen representatives $H, H'$). It turns out that minimum representatives of an equivalence class $[H]$ correspond to shortest paths from $[\emptyset]$ to $[H]$ in that graph. Thus, beside having the weakness of a significant memory overhead because all visited classes have to be kept in memory in order to guarantee a non-redundant traversal, this strategy has two major advantages:

1. it generates minimum representatives of each equivalence class without additional cost and,
2. as it visits equivalence classes in ascending order with respect to the minimum cardinality of their members, it allows for pruning based on a length-limit.

The straightforward implementation of that graph traversal has the serious drawback that it reaches vertices via many redundant ways: a minimum representative $G$ induces $|G|!$ different paths to $[G]$—one for each of its orderings. This effect can be significantly reduced by chosing some arbitrary but fixed order $\{c_1, \ldots, c_N\}$ of the constraint set $\boldsymbol{C}$. Thereby the expressions "$\max H$" and "$\min H$" are defined for non-empty descriptions $H \in \mathcal{L}$ by referring to the constraint in $H$ with the maximum or minimum index, respectively. Algorithm 1 below uses this order to reduce its traversal paths to those that are in descending order. To describe its behavior in more detail and prove its correctness we define a modified **lexicographical order** on the descriptions $\mathcal{L}$, denoted by "$\prec_L$", given by:

$$H \prec_L H' \Leftrightarrow |H| < |H'| \vee (|H| = |H'| \wedge \max(H \Delta H') \in H') \ .$$

Using this strict linear order we can specify the elements that Algorithm 1 enumerates among the potentially many minimum representatives.

**Definition 2 (Canonical Minimum Representative).** *The **canonical minimum representative** of an equivalence class $[H]$, denoted by $\mu(H)$, is the unique minimum representative of $[H]$ that is minimal with respect to $\prec_L$.*

These canonical minimum representatives have the important property that they can be built from one another inductively via their suffixes.

**Lemma 8** *Let $G \neq \emptyset$ be a non-empty canonical minimum representative of its equivalence class $[G]$, i.e., $G = \mu(G)$. Then $G' = G \setminus \{\min G\}$ is the canonical minimum representative of $[G']$.*

*Proof.* Assume there is a $G'' \in [G']$ with $G'' \prec_L G'$. Then $G'' \cup \{\min G\} \prec_L G' \cup \{\min G\} = G$. But as

$$\begin{aligned} \mathcal{D}[G'' \cup \{\min G\}] &= \mathcal{D}[G''] \cap \mathcal{D}[\{\min G\}] \\ &= \mathcal{D}[G'] \cap \mathcal{D}[\{\min G\}] \\ &= \mathcal{D}[G' \cup \{\min G\}] = \mathcal{D}[G] \ , \end{aligned}$$

i.e., $(G'' \cup \{\min G\}) \in [G]$, this contradicts $G = \mu(G)$. □

Now we can prove the correctness and time complexity of Algorithm 1.

**Theorem 9** *Algorithm 1 exactly and non-redundantly lists $\mu(H)$ for all $[H] \in \mathcal{R}^{\equiv} = \{[H] : q(H) \geq q^*\}$ in lexicographical order in time $O(|\mathcal{S}^{\equiv}|\, n^2 m^2)$ and space $O(|\mathcal{S}^{\equiv}|\, nm)$ where $\mathcal{S}^{\equiv} = \{[H] : \hat{q}(H) \geq q^*\}$ is the search space of all potentially interesting equivalence classes.*

*Proof.* For each dequeued tuple at most $|\boldsymbol{C}|$ augmentations are evaluated involving a computation of $\sigma$ and a visited check. The prefix-tree lookup is performed in time $|S| \leq |\boldsymbol{C}|$ and $\sigma$ is computed in time $nm$. Also the space dominant data structure $\mathcal{V}$ contains at most one element of size at most $|\boldsymbol{C}|$ for each dequeued tuple. As $|\boldsymbol{C}|$ is bounded by $nm$ the claim follows if we can show that in lexicographical order for each $[H] \in \mathcal{S}^{\equiv}$ a tuple $(\mu(H), \sigma(H), \boldsymbol{A})$ is enqueued and only tuples of this form are enqueued, i.e., if the following three properties hold:

**Algorithm 1** Inductive Minimum Representative Construction

Require: ordered ground set of constraints $\boldsymbol{C} = \{c_1, \ldots, c_N\}$,
extension closure operator $\sigma$,
quality function $q$ with optimistic estimator $\hat{q}$, and quality threshold $q^*$
Output: family $\{\mu(H)\colon q(H) \geq q^*\}$ in lexicographical order

1. **init** $\mathcal{Q}$ as empty queue and $\mathcal{V}$ as empty prefix tree
2. **enqueue** $(\emptyset, \sigma(\emptyset), \boldsymbol{C})$ on $\mathcal{Q}$
3. **while** $\mathcal{Q} \neq \emptyset$ **do**
4. **dequeue** front element $(G, S, \boldsymbol{A})$ of $\mathcal{Q}$
5. **if** $q(S) \geq q^*$ **then print** $G$
6. $\boldsymbol{A'} \leftarrow \{c \in \boldsymbol{A} \setminus S\colon\ c < \min G,\ \hat{q}(G \cup \{c\}) \geq q^*\}$
7. **for all** $c_i \in \boldsymbol{A'}$ in ascending order of their index **do**
8. $G' \leftarrow G \cup \{c_i\}$
9. $S' \leftarrow \sigma(G')$
10. **if** $S' \notin \mathcal{V}$ **then**
11. **add** $S'$ to $\mathcal{V}$
12. **enqueue** $(G', S', \boldsymbol{A}')$ on $\mathcal{Q}$

(i) If a tuple $(G, S, \boldsymbol{A})$ is enqueued before $(G', S', \boldsymbol{A'})$ then $G \prec_L G'$.
(ii) For all $[H] \in \mathcal{S}^{\equiv}$ a tuple $(G, S, \boldsymbol{A})$ with $G = \mu(H)$ and $S = \sigma(H)$ is enqueued, and $\boldsymbol{A} \supseteq \{c \in \boldsymbol{C}\colon\ (G \cup \{c\}) \in \mathcal{S}^{\equiv},\ c < \min G\}$.
(iii) all enqueued tuples $(G, S, \boldsymbol{A})$ are of the form $G = \mu(H)$ and $S = \sigma(H)$ for some $[H] \in \mathcal{S}^{\equiv}$.

Property (i) is implied by the breadth-first strategy and the following observation: if $G_1 \prec_L G_2$ then all descriptions $G_1'$ generated from $G_1$ are lexicographically smaller than all descriptions $G_2'$ generated from $G_2$.

Assume that (ii) is violated for some $[H]$. Then choose a class $[H]$ that violates (ii) with a minimal $G' = \mu(H)$. As $(\emptyset, \sigma(\emptyset), \boldsymbol{C})$ is enqueued in line 2, it holds that $G' \neq \emptyset$. By Lemma 8, $G = G' \setminus \{\min G'\}$ is lexicographically minimal in $[G]$. The anti-monotinicity of the search space and $G \subset G'$ imply that (ii) holds for $[G]$. In particular a tuple $(G, S, \boldsymbol{A})$ is enqueued with $(\min G') \in \boldsymbol{A}$ because $\min G' < \min G$ (for the same reason and because of the anti-monotinicity of $\mathcal{S}^{\equiv}$, the augmentation set $\boldsymbol{A'}$ satisfies $\boldsymbol{A'} \supseteq \{c \in \boldsymbol{C}\colon\ (G' \cup \{c\}) \in \mathcal{S}^{\equiv},\ c < \min G'\}$). Thus, $G'$ is generated subsequently in line 8. Then $\sigma(G')$ does not pass the visited check in line 10. This implies that $[G']$ has already been visited, say via $G'' \in [G']$. It follows from $(i)$ that $G'' \prec_L G'$ contradicting $G' = \mu(G')$.

For (iii) observe that $S = \sigma(G)$ for all enqueued tuples by the generation of $S$ in line 9. Now assume that $G \neq \mu(S)$ for an enqueued tuple $(G, S, \boldsymbol{A})$. Then there is an $G' \in [S]$ with $G' \prec_L G$. By the anti-monotinicity of the search space and (ii) a tuple $(G', S', \boldsymbol{A'})$ is enqueued, and by (i) it is enqueued before $(G, S, \boldsymbol{A})$. In the same iteration $S' = S$ is added $\mathcal{V}$. Consequently, $(G, S, \boldsymbol{A})$ can not be enqueued as it does not pass the visited check in line 10—a contradiction. □

There are some additional speedups that do not affect the worst-case time complexity. For the top-$k$ scenario the same changes as for the closure/greedy approach can be applied. Furthermore, if at a node $(G, S, \boldsymbol{A})$ with $c_i, c_j \notin S$ it holds that $c_j \in \sigma(G \cup \{c_i\})$ then it follows $\sigma(G \cup \{c_i, c_j\}) = \sigma(G \cup \{c_i\})$ by monotonicity of $\sigma$. In this case the augmentation element $c_i$ can be removed from $\boldsymbol{A'}$ of the child $(G \cup \{c_j\}, \sigma(G \cup \{c_j\}), \boldsymbol{A'})$ in case $c_i \prec_L c_j$ as it would redundantly generate the same equivalence class again. Furthermore, the sorting of the constraint set can have a substantial impact on the computation time. It is, however, a non-trivial problem to find an optimal sorting (see [9] for a comparison of different sorting strategies for divide and conquer closed set listing).

## 7 Empirical Evaluation

In this section we empirically compare non-redundant subgroup discovery with minimum representatives to traditional subgroup discovery. This includes an evaluation of both proposed algorithmic solutions for Problem 1. We considered

| dataset | credi. | lung-. | lymph | mush. | nurse. | sick | soybe. | splice | tic-t. | vote |
|---|---|---|---|---|---|---|---|---|---|---|
| label | bad | 1 | maln. | pois. | recm. | sick | EI | bspot. | pos. | repub. |
| $n$ | 15 | 56 | 18 | 22 | 8 | 39 | 60 | 35 | 9 | 16 |
| $m$ | 1000 | 32 | 148 | 8124 | 12960 | 3772 | 3190 | 638 | 958 | 435 |
| $\lvert\boldsymbol{C}\rvert$ | 58 | 159 | 50 | 117 | 27 | 66 | 133 | 287 | 27 | 48 |

**Table 3.** Datasets

ten datasets from the UCI Machine Learning Repository [1], which are presented along with their most important properties in Table 3. All numerical attributes where discretized using minimal entropy discretization. As representative traditional subgroup miner we used the state-of-the-art algorithm Dpsubgroup [10]. All involved algorithms were implemented in Java and will be published on the author's webpage. For the sake of a better comparision we used a simplified reimplementation of the LCM algorithm and not the implementation published by its author. The quality function was the binomial test quality function combined with the optimistic estimator introduced in Section 2. All experiments were performed on a Core 2 Duo E8400 @ 3Ghz running a Sun SE 6u10 Java Virtual Machine with 1.5 GB of Java heap space under Windows XP.

*Computation Time* Table 4 contains the computation times that correspond to the compression experiments presented in Section 3 for Dpsubgroup (dpsg), LCM/greedy (lcm/gr), and Algorithm 1 (imr). The threshold $t100$, i.e., the quality of the 100th best subgroup description, is explicitly stated. The results essentially reflect the already observed search space reduction. Although, even for datasets on which a compression is achieved traditional subgroup discovery is not directly outperformed. This is illustrated in Figure 1 in which the development of search spaces and computation time for increasing length-limits are

| dataset | credi. | lung-. | lymph | mush. | nurse. | sick | soybe. | splice | tic-t. | vote |
|---|---|---|---|---|---|---|---|---|---|---|
| $q^*$ | 0.094 | 0.336 | 0.244 | 0.267 | 0.029 | 0.177 | 0.223 | 0.190 | 0.061 | 0.306 |
| dpsg | **2** | 84.4m | 0.5 | **0.6** | **1.2** | 4.3h | 10.6h | 23 | 0.3 | 0.4 |
| lcm/gr | 3.2 | **23** | 0.3 | 1.0 | 2.3 | **18.3m** | 123 | 38 | **0.2** | **0.2** |
| imr | 3.6 | **23** | **0.2** | 0.9 | 2.4 | oom | **115** | **20** | **0.2** | 0.3 |
| $q^*$ | $\epsilon$ | | | | | | | | | |
| dpsg | 242 | >12h | 457 | >12h | **2** | >12h | >12h | >12h | **1** | 127 |
| lcm/gr | 184 | 95 | 6.5 | 53m | 85.5 | **7h** | **41m** | >12h | 15 | 59 |
| imr | **26** | **60** | **4** | **39** | 5 | oom | oom | oom | 2 | **19** |

**Table 4.** Computation time (in seconds unless stated differently); "oom" and ">12h" for computations that ran out of memory or out of time, respectively

shown. The Dpsubgroup algorithm is only beaten by Algorithm 1 for sufficiently large differences in the search space. This behavior is due to the sophisticated data structures (fptrees [2, 11]) Dpsubgroup uses in contrast to our algorithm.

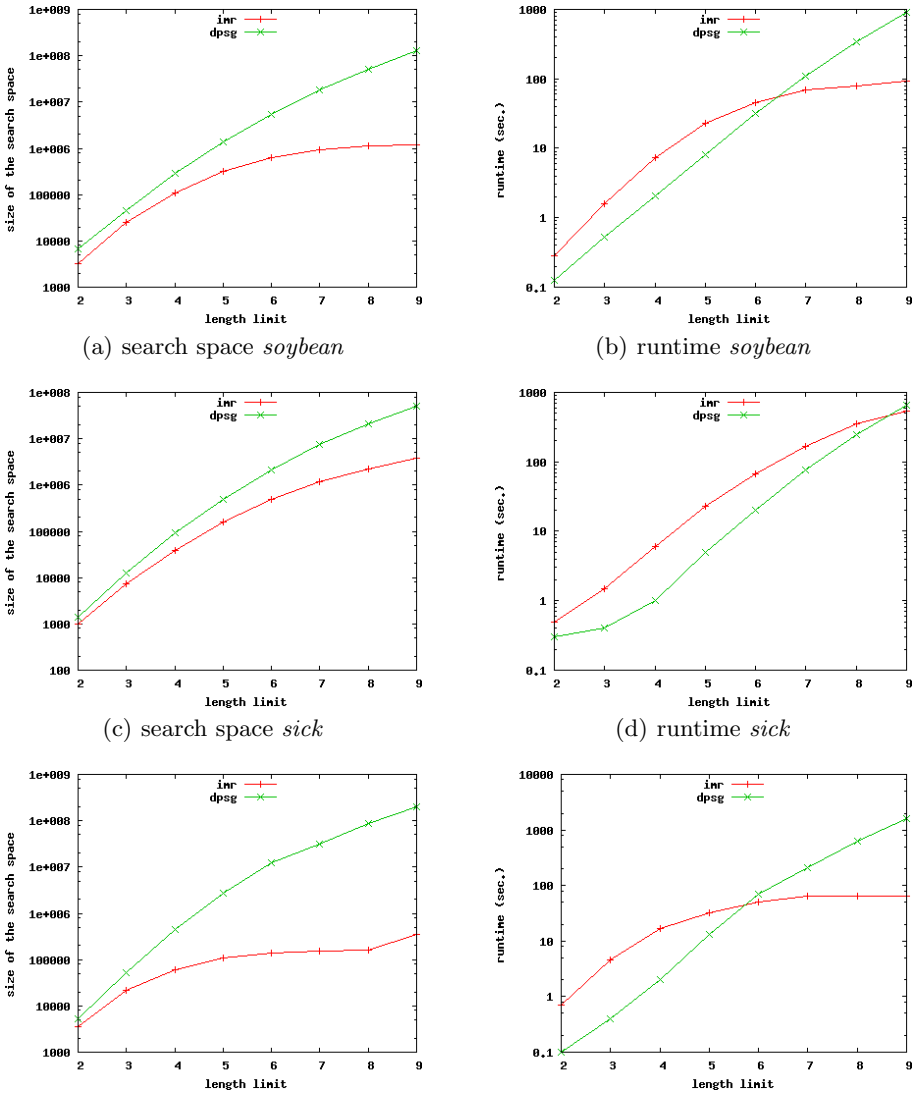
A further noteworthy fact is that unless Algorithm 1 ran out of memory (the oom entries) it always outperforms LCM/greedy. This motivates a more detailed investigation of the latter approach.

*Greedy Performance* We analyze the perfomance of the greedy algorithm within the lcm/greedy approach in two respects: (a) the length of the produced representatives and (b) the greedy algorithm's fraction of the computation time for the experiments with $q^* = \epsilon$. Note that this is an extremely unfavorable case for the lcm/greedy approach because of the large number of interesting equivalence classes, each of which requires one greedy call. The results are listed in Table 5. We note that (a) for all datasets the length of the subgroup descriptions obtained using the greedy algorithm is only marginally greater than the minimum length and (b) the computation time of LCM/greedy was dominated by the greedy algorithm. Without the greedy approximations LCM even slightly outperformed Algorithm 1 on most datasets.

*Predictive Power* While it is out of scope of this article to evaluate the claim that selecting minimum representatives improves understandability for users, their power as building blocks of global prediction models can be supported with standard accuracy experiments. Although not their primary intention, subgroup description families $\mathcal{R}$ are sometimes evaluated (see [13]) by investigating the area under the ROC curve (AUC) of the set of global models $\{h_i \colon 1 \leq i \leq |\mathcal{R}|\}$,

| | credi. | lung-. | lymph | mush. | nurse. | sick | soybe. | splice | tic-t. | vote |
|---|---|---|---|---|---|---|---|---|---|---|
| avg. min. | 5.588 | 4.914 | 4.532 | 5.776 | 5.328 | ? | ? | ? | 5.005 | 5.991 |
| avg. apx. | 5.603 | 5.032 | 4.562 | 5.862 | 5.328 | 9.062 | 6.591 | ? | 5.041 | 6.039 |
| max. dif. | 5 vs. 3 | 7 vs. 4 | 5 vs. 3 | 6 vs. 4 | no diff | ? | ? | ? | 6 vs. 4 | 8 vs. 5 |
| time frac. | 0.9 | 0.07 | 0.54 | 0.99 | 0.97 | 0.96 | 0.78 | ? | 0.93 | 0.59 |

**Table 5.** Greedy performance for $q^* = \epsilon$




(a) search space *soybean* (b) runtime *soybean*

(c) search space *sick* (d) runtime *sick*

(e) search space *lung-*. (f) runtime *lung-*.

**Fig. 1.** Logscale search space and runtime comparison of Algorithm 1 (imr) vs. Dp-subgroup (dpsg) for increasing length limits

| dataset | credi. | lung-. | lymph | mush. | nurse. | sick | splice | soybe. | tic-t. | vote |
|---|---|---|---|---|---|---|---|---|---|---|
| RIPPER | 0.619 | 0.729 | 0.774 | **1.0** | **0.815** | 0.916 | 0.969 | **0.917** | 0.975 | **0.959** |
| sgd | 0.628 | 0.708 | 0.757 | 0.890 | 0.813 | 0.908 | **0.987** | 0.792 | **0.999** | 0.972 |
| min-repr | **0.633** | **0.731** | **0.831** | 0.946 | 0.813 | **0.928** | **0.987** | 0.856 | **0.999** | 0.972 |

**Table 6.** Average AUC over ten 5-fold cross-validations

where $h_i$ classifies a given data point as positive if it supports any of the $i$ highest quality subgroup descriptions from $\mathcal{R}$. For each dataset we compared the predictive quality of the

- top-20 subgroup description (sgd),
- minimum representatives of the top-20 equivalence classes (min-repr),
- and additionally the hypothesis of the rule learner J-RIP, a Java implementation of RIPPER [4].

A rule learner was chosen as additional benchmark because, among supervised learning methods, the nature of the hypotheses it produces is most similar to the subgroup based models. Table 6 shows the average AUC of ten cross validations using five folds. For performance reasons, we executed all subgroup discoveries with a depth limit of 5. Using equivalence classes instead of subgroup descriptions always resulted in a higher or equal AUC. Perhaps surprisingly, in addition, the hypotheses build from the minimum representatives also outperformed the rule learner hypotheses on the majority of datasets.

## 8 Conclusion

*Discussion* Beside the results stated in the introduction our experiments primarily revealed the following trade-off in using non-redundant subgroup discovery based on equivalence classes: while one gains a significant compression of the search space and output, this is sometimes outweighted by the fact that one loses the sophisticated data structures of traditional methods. There are, however, many datasets/quality thresholds managable by our algorithms, that were completely intractable before due to an exponential explosion of the search space. This opens up new opportunities for some datasets like an exhaustive enumeration of all positive quality equivalence classes with a subsequent global optimization step.

*Future Work* It is important to note that the results of this article easily generalize to *more expressive constraint languages* as long as they guarantee unique maximal representatives and extension anti-monotonicity with respect to some specialization relation. This is for instance the case for interval constraints, which are appealing in the presence of ordinal attributes. In constrast to the standard attribute/value equality constraints of the form "$A_i = v$" an intervall constraint for a real-valued attribute is of the form "$A_i \in [l, u]$". In the presence of more expressive constraint languages the equivalence class search space becomes even

more important as there is an increasing number of ways to (redundantly) describe one and the same extension. This motivates a future study investigating the combination of non-redundant subgroup discovery and other constraint languages.

## Acknowledgment

We thank the anonymous reviewers for their helpful comments. This research was partially supported by the European Commission under the project *RACWeB* (No. 045101).

## References

1. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
2. Martin Atzmüller and Frank Puppe. SD-map - a fast algorithm for exhaustive subgroup discovery. In *PKDD*, pages 6–17, 2006.
3. Toon Calders, Christophe Rigotti, and Jean franois Boulicaut. A survey on condensed representations for frequent sets. In *Constraint Based Mining and Inductive Databases*, pages 64–80. Springer, 2005.
4. William W. Cohen. Fast effective rule induction. In *ICML*, pages 115–123, 1995.
5. Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
6. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer Verlag, 1999.
7. Gemma C. Garriga, Petra Kralj, and Nada Lavrač. Closed sets for labeled data. *J. Mach. Learn. Res.*, 9:559–580, 2008.
8. Friedrich Gebhardt. Choosing among competing generalizations. *Knowledge Acquisition*, 3(4):361 – 380, 1991.
9. Alain Gély. A generic algorithm for generating closed sets of a binary relation. *Formal Concept Analysis*, pages 223–234, 2005.
10. Henrik Grosskreutz, Stefan Rüping, and Stefan Wrobel. Tight optimistic estimates for fast subgroup discovery. In *ECML/PKDD (1)*, pages 440–456, 2008.
11. Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *SIGMOD Conference*, pages 1–12, 2000.
12. W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI, 1996.
13. N. Lavrac, B. Kavsek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5(Feb):153–188, February 2004.
14. Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Inf. Syst.*, 24(1):25 – 46, 1999.
15. Petr Slavík. A tight analysis of the greedy algorithm for set cover. *Journal of Algorithms*, 25(2):237–254, 1997.
16. Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Discovery Science*, pages 16–31, 2004.
17. Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *PKDD*, pages 78–87. Springer, 1997.