Towards Polynomial Subgroup Discovery by means of FCA[⋆]

Aleksey Buzmakov[0000−0002−9317−8785]

National Research University Higher School of Economics, Russia
avbuzmakov@hse.ru

Abstract. The goal of subgroup discovery is to find groups of objects that are significantly different than "average" object w.r.t. some supervised information. It is a computational intensive procedure that traverses a large searching space corresponding to the set of formal concepts. It was recently found that a part of formal concepts, called stable concepts, can be found in polynomial time. Accordingly, in this paper a new algorithm, called `SD-SOFIA`, is presented. `SD-SOFIA` fits subgroup discovery process in the framework of stable concept search. The proposed algorithm is evaluated on a dataset from UCI repository. It is shown that its practical computational complexity is polynomial.

Keywords: Subgroup Discovery · Stable Concepts · Supervised Learning · Exploratory Data Analysis · Algorithms.

Introduction

Subgroup discovery is supervised data mining technique allowing for finding groups of objects that express unexpected behavior w.r.t. some supervised information [1]. For example, class labels of objects is an example of such supervised information. Subgroup discovery not only enumerates the objects with unexpected behavior but also describes them in a human readable form providing a way for understanding the connection between the supervised information and the description space. Such understanding is important for analysis of the real process generating the supervised information.

Formal concept analysis (FCA) [11] is a well-established mathematical formalism suitable for description of subgroup discovery process. Indeed, extents of formal concepts are subgroups, their intents are subgroup descriptions. Then every concept can be evaluated by means of a quality function that relates object class labels[1] and the concept interest. Then the task of the subgroup discovery in these terms is to find the concept with the best value of the quality function.

One of the challenges in subgroup discovery is the exponentially large searching space of concepts that entails two consequences. First, it is computationally hard to find the best subgroups. Second, since the searching space is large, it is

⋆ The reported study was funded by RFBR, project number 19-31-37001

1 For simplicity only classification task is considered.

likely that some its elements are associated with high values of quality function just by chance, i.e., spurious findings are possible. There are a number of approaches that deals with these consequences. In particular, the searching space can be limited, e.g., it can be stated that only concepts with few attributes constitute the searching space. Another possible way is to rely on a certain heuristic such that it is unlikely to miss the best subgroups [14, 5]. Finally, some approaches allow gradually refining the searching space in such a way that their computation can be stopped at any moment and the best found result is reported [2].

The problem of spurious findings can be solved by controlling the statistical significance of the examined subgroups [15]. Such approaches can be naturally combined with the approaches that limit the searching space. By contrast, combing statistically significant subgroup discovery with heuristic methods is hard.

Accordingly, in this paper we discuss a limitation of the searching space that is based on stability of a formal concept. This choice is motivated by two facts. First, stability is a meaningful concept selection method. Indeed, stability is the probability of a concept to be preserved after random deletion of objects from the dataset. Thus, if just a small modification of the object set is enough for removing a concept, then probably this concept can be removed from the searching space. Second, it was recently shown that stability threshold can be adjusted on the fly, allowing for polynomial time computational procedure [8, 6]. Consequently, combining this procedure with subgroup discovery give an opportunity for controling the subgroup discovery computation time and allowing for statistically significant subgroup discovery.

Accordingly, the main contribution of this paper is the algorithm combining the subgroup discovery process and the stability threshold adjustment allowing for practical polynomial time complexity subgroup discovery with mathematical guarantees for the resulting subgroup.

The rest of the paper is organized as follows. Section 1 provides some basic definitions, then in Section 2 the task is defined. Later the algorithms proposed in this paper are discussed in Section 3. Finally, before concluding the paper the algorithms are evaluated an a dataset from the UCI repository [9].

1 Definitions

Subgroup discovery task statement depends (1) on the searching space and (2) on the quality function. The searching space is defined by means of FCA, while quality function is considered to be known and is not discussed in this paper.

1.1 Formal concept analysis

For simplicity the dataset is described as a formal context (G, M, I), where G is the set of objects, M is the set of attributes, and $I \subseteq G \times M$ is a relation between. The sets of objects and attributes are connected by means of a derivation operator[2]:

$$A^{\uparrow} = \{B \subseteq M \mid \forall g \in A\,((g,m) \in I)\}, \text{ for } A \subseteq G, \tag{1}$$

$$B^{\downarrow} = \{A \subseteq G \mid \forall m \in B\,((g,m) \in I)\}, \text{ for } B \subseteq M. \tag{2}$$

A formal concept is a pair (A, B), where $A \subseteq G$ is callede extent and $B \subseteq M$ is called intent, such that $A = B^{\downarrow}$ and $B = A^{\uparrow}$. The set of concepts is ordered w.r.t. the order on extents (or dually inverse order on intents). This order is a lattice called concept lattice.

1.2 Projections

The algorithm presented in Section 3 is based on the notion of projections. Originally, it was introduced within the framework of pattern structures [10]. However, for the sake of simplicity, it is presented here in terms of standard FCA.

Definition 1. *Projection $\psi : 2^M \to 2^M$ is a function sutisfying:*

- *idempotency, i.e., $\psi(\psi(X)) = \psi(X)$;*
- *monotonicity, i.e., $X \subset Y \longrightarrow \psi(X) \subseteq \psi(Y)$;*
- *contractivity, i.e., $X \supseteq \psi(X)$.*

In this paper, a special kind of projections is considered corresponding to removal of certain attributes. In particular, $\psi_Y(X) = X \cap Y$, i.e., it removes all attributes outside of Y. It can be seen, that the ψ_Y is a projection. The larger the set Y the more attributes are preserved after the projection.

As it will be discussed later the algorithm starts from the most simple projections, removing many attributes, and then iteratively adds attributes one by one updating the result. A similar approach was used in [2] where numerical intervals were iteratively updated.

1.3 Subgroup Discovery

From the subgroup discovery (SD) point of view the concept lattice is the searching space. The extent of a formal concept is the subgroup and the intent of a formal context is the description of this subgroup. Given a quality function $Q : 2^G \to \mathbb{R}$, the goal of SD is to find the formal concept with extent maximizing the quality function Q.

Let us consider a standard quality function for the classification task [13]. Let class labels of objects are given by a function $\texttt{class} : G \to \{0, 1\}$, where $\{0, 1\}$ is the set of available class labels. Given a set of objects A, the weighted relative accuracy can be expressed as

$$Q_1(A) = \frac{|A|}{|G|} \cdot \left(\frac{|\{g \in A \mid \texttt{class}(g) = 1\}|}{|A|} - \frac{|\{g \in G \mid \texttt{class}(g) = 1\}|}{|G|} \right).$$

[2] The operators $(\cdot)^{\downarrow}$ and $(\cdot)^{\uparrow}$ are used instead of the standard $(\cdot)'$, since it makes the reading more clear w.r.t. the operator argument.

This quality function express a trade-off between the size of the subgroup $\frac{|A|}{|G|}$ and the improvement in the "purity" of class labels within the concept w.r.t. the average "purity".

Although it is possible to iterate over all formal concepts and compute their quality, it is not efficient. Thus, a typical exhaustive subgroup discovery procedure is implemented as a branch-and-bound search. It is based on the following two ingredients [3]:

- A refinement operator $\mathtt{r} : 2^G \to 2^{2^G}$, that is monotone, i.e., given $B_0 \subseteq M$, $(\forall B \in \mathtt{r}(B_0)) B \subseteq B_0$. The refinement operator organizes the procedure of generating new "more specific" candidate subgroups based on already found ones.
- An optimistic estimator $\overline{Q} : 2^G \to \mathbb{R}$ of the subgroup quality function Q. The optimistic estimator $\overline{Q}(A)$ is the upper bound of the quality function Q on any subset of A, i.e., $(\forall S \subseteq A) \overline{Q}(A) \geq Q(S)$.

Given these two components and the quality of the already found best subgroup q_{best} the discovering procedure is as follows. For any found subgroup A with description $B_0 = A^{\uparrow}$ one can verify if any subset of A is a potentially interesting subgroup, i.e., $\overline{Q}(A) > q_{best}$. If not the subgroup A can be ignored, otherwise it is refined by means of the refinement operator $\mathtt{r}$. The refined subgroups with description $B \in r(B_0)$ are evaluated by means of the subgroup quality function Q and if any subgroup is better then the already found one, the best subgroup is updated.

1.4 Formal Concept Stability and Δ-stability

The branch and bound computational efficiency is still limited if the dataset is large. The further improvement of the efficiency can be made by modification of the searching space. One way is to consider only "stable" concepts as the searching space.

Stability of a formal concept [12] measures how strong the concept depend on the dataset. If removal of random objects from the dataset is likely to remove a concept then the concept is not stable.

It was shown that, given a context and a concept, the computation of concept stability is #P-complete [12]. Accordingly, in [7] bounds on stability of a concept was discussed, in particular, stability is bounded as follows:

$$1 - \sum_{d \in \mathrm{DD}(c)} \frac{1}{2^{\Delta(c,d)}} \leq \mathrm{Stab}(c) \leq 1 - \max_{d \in \mathrm{DD}(c)} \frac{1}{2^{\Delta(c,d)}}, \tag{3}$$

where $\mathrm{DD}(c)$ is the set of all direct descendants of a concept c in the lattice and $\Delta(c,d)$ is the size of the set-difference between extent of c and extent of d, i.e. $\Delta(c,d) = |\operatorname{Ext}(c) \setminus \operatorname{Ext}(d)|$. It can be seen that stability in 3 is bounded tightly if $\Delta(c) = \min\limits_{d \in \mathrm{DD}(c)} \Delta(c,d)$ is high. Moreover, the higher the stability is, the tighter the bounds. Thus, in order to identify the most stable concepts one can switch to $\Delta(c)$, called Δ-measure, which is polynomially computable.

Recently, it was also shown, that given a Δ-measure threshold θ, it is possible to directly find formal concepts with Δ-measure higher than θ [8]. Moreover, introducing a special adjustment procedure, one is able to extract concepts with the highest value of Δ-measure in polynomial time [6]. Accordingly, in this paper this approach is translated to subgroup discovery task.

2 Task Statement

The goal of this paper is to present an algorithm for subgroup discovery task modifying the searching space in such a way that its practical computational complexity is polynomial.

More formally, given a formal context $\mathbb{K} = (G, M, I)$ and a quality function Q of a formal concept, the goal is an algorithm that finds the threshold θ of Δ-measure and the corresponding best Δ-stable concept $\mathcal{C}$, w.r.t. Q, such that the practical computational complexity is polynomial w.r.t. $\mathbb{K}$, $\mathbb{T}(Q)$, and the memory usage limit L, where $\mathbb{T}(Q)$ is the computational complexity of Q, and L is the maximal number of potential concepts to be extended at the same time.

3 Algorithms

3.1 Algorithm ϑ-Σοφια

Let us first remind the original algorithm Σοφια [8, 6]. It is based on the following observations.

Proposition 1 (Projection antimonotonicity). *Given a context $\mathbb{K} = (G, M, I)$, for any projection $\psi : 2^M \to 2^M$ and a concept $\mathcal{C} = (A, B)$ it is valid that*

$$\Delta_{(G,M,I)}(\mathcal{C}) \leq \Delta_{(G,\psi(M),I)}\left((\psi(B)^{\downarrow}, \psi(B))\right). \tag{4}$$

It should be noticed, that Δ-measure depends on the structure of the concept lattice, i.e., Δ-measure should be computed when the lattice is fixed. Accordingly, in (4) every Δ-measure is indexed with the corresponding formal context. It also should be noticed that $(\psi(B)^{\downarrow}, \psi(B))$ is indeed a formal concept in $(G, \psi(M), I)$ as it is shown earlier [10].

This property (4) gives a way for direct search for Δ-stable concepts. Indeed, a concept $\mathcal{C} = (A, B)$ is not Δ-stable in $(G, \psi(M), I)$ for some ψ, then any preimages of B cannot be the intents of Δ-stable concepts in (G, M, I), i.e., any concept with intent $\hat{B}$, such that $\psi(\hat{B}) = B$ is not Δ-stable. Thus, if one is able to find Δ-stable concepts in $(G, \psi(M), I)$, then only the preimages of these concepts w.r.t. ψ can be Δ-stable in (G, M, I).

However, how can one efficiently find Δ-stable concepts in $(G, \psi(M), I)$? Since $(G, \psi(M), I)$ is a context it is possible to consider a projection of it. It forms a chain of projections and Δ-stable concepts are first found in the most

```
   Data: A context $\mathbb{K} = (G, M, I)$, a chain of projections $\Psi = \{\psi_0, \psi_1, \cdots, \psi_k\}$,
         and a threshold $\theta$ for $\Delta$-measure.
1  Function ExtendProjection($i$, $\theta$, $\mathcal{P}_{i-1}$)
       Data: Projection number $i$, a threshold value $\theta$, and the set of concepts
             $\mathcal{P}_{i-1}$ for the projection $\psi_{i-1}$.
       Result: The set $\mathcal{P}_i$ of all concepts with the value of $\Delta$-measure higher
               than the threshold $\theta$ for the projection $\psi_i$.
2      $\mathcal{P}_i \longleftarrow \emptyset$;
3      /* Put all preimages in $\psi_i(\mathbb{K})$ for any concept $p$                */
4      foreach $p \in \mathcal{P}_{i-1}$ do
5          $\mathcal{P}_i \longleftarrow \mathcal{P}_i \cup \text{Preimages}(i, p)$
6      /* Filter concepts in $\mathcal{P}_i$ to have a value of Δ higher than $\theta$   */
7      foreach $p \in \mathcal{P}_i$ do
8          if $\Delta_{\psi_i}(p) \le \theta$ then
9              $\mathcal{P}_i \longleftarrow \mathcal{P}_i \setminus \{p\}$
10 Function $\vartheta$-$\Sigma o\varphi\iota\alpha$
       Result: The set $\mathcal{P}$ of all concepts with a value of $\Delta$ higher than $\theta$.
11     $\mathcal{P} \longleftarrow (\emptyset^{\downarrow}, \emptyset)$;
12     /* Run through out the chain $\Psi$                                            */
13     foreach $0 < i \le |M|$ do
14         $\mathcal{P} \longleftarrow$ ExtendProjection($i, \theta, \mathcal{P}$);
```

Algorithm 1: The Θ-Σοφια algorithm for finding concepts in $\mathbb{K}$ with a value of a Δ-measure higher than a threshold θ.

general projections removing all attributes, then the next projections removes all but one attribute, the next one removes all but two attributes, on so on.

This procedure is shown in Algorithm 1 and is called ϑ-Σοφια algorithm. The computational complexity of this algorithm is

$$O\left(|M| \cdot \max_{0<i\le|M|} |\mathcal{P}_i| \cdot (\mathbb{T}(\text{Preimages}) + \mathbb{T}(\Delta))\right).$$

It become polynomial if the threshold θ is adjusted in such a way that $|\mathcal{P}_i| < L$ for any i and a predefined memory limit L.

3.2 Algorithm `SD-SOFIA`

Usage of Algorithm 1 is limited for subgroup discovery, since it finds preimages of all concepts from projection i to projection $i+1$ simultaneously. By contrast, the most commonly used strategy in subgroup discovery is expansion of the most promising concept [4]. This allows for earlier finding of concepts with high quality Q improving the efficiency of branch cutting.

It should be noticed that in Algorithm 1 finding preimages of a concept does not depend on other concepts and, thus, concepts that are stored in $\mathcal{P}$ can correspond to different projections. In this case, one is able to choose the order of concept expansion and, in particular, it can be used for expanding first

```
Function FindBestConcept()
    queue.Push $((G, \perp \mid 0))$;                    /* Projection number 0 */
    while not queue.isEmpty() do
        $c \leftarrow$ queue.PopTheMostPromissing ();
        $\{c_i\} \leftarrow$ Preimages($c$);
        foreach $cc \in \{c_i\}$ do
            if $\mathrm{Proj}(cc) = |M|$ then
                best.Register($cc$);
                next;
            if $\Delta_{\mathrm{Proj}(cc)}(cc) < \theta$ then
                next;
            if not best.IsPromissing($cc$) then
                next;
            queue.Push($cc$);
        if queue.Size() $> L$ then
            $\theta \leftarrow$ AdjustThld(queue, $\theta$);
```
Algorithm 2: The `SD-SOFIA` algorithm identifying the Δ-measure threshold θ and the corresponding best concept w.r.t. a quality function Q. The algorithm ensures the polynomial computational complexity.

the most promising concepts w.r.t. subgroup discovery task. This procedure is shown in Algorithm 2. All concepts are stored in a `queue`. The `queue` can contain concepts from different projections, thus, a concept is denoted as $(A, B \mid i)$, where A and B are the extent and the intent of the concept correspondingly, and i is the projection it is computed in. The corresponding elements of a concept $c = (A, B \mid i)$ can be extracted by means of functions Ext, Int, and Proj for the extent, the intent, and the projection number of c correspondingly.

Let us first fix some order on attributes M. Let M_i be the first i attributes from M. Then, this algorithm relies on the following chain of projections $\Psi = \langle \psi_0, \psi_1, \ldots, \psi_{|M|} \rangle$, where $\psi_i(X) = X \cap M_i$, i.e., it removes all attributes but the first i attributes from M. In line 2, Algorithm `SD-SOFIA` initializes the `queue` with the only available concept in projection 0. This concept is $(G, \emptyset)$. Indeed, since M_0 contain no attribute, the only available intent is $\emptyset$.

Then, while `queue` is not empty the concepts are extracted one by one. For subgroup discovery task the concepts are extracted (line 4) w.r.t. their potential, i.e., the value of the optimistic estimate $\overline{Q}$ of the quality function Q. Then in line 5 the preimages of the most promissing concept $c = (A, B \mid i)$ are computed. Since projection ψ_{i+1} preserves one more attribute than projection ψ_i, there are only two possible preimages: $c_1 = (A, B^{\downarrow\uparrow} \mid i + 1)$ and $c_2 = ((B \cup \{i\})^{\downarrow}, (B \cup \{i\})^{\downarrow\uparrow} \mid i + 1)$. The preimages c_1 and c_2 can coincide and, thus, in this case only one of them should be considered.

Then every preimage cc of the concept c is processed in lines 7–14. First in lines 7–9 it is verified that cc is already in the last projection $\psi_{|M|}$. Only in this case the final Δ-measure value for this concept is known. Thus, only in this moment it is possible decide if this concept can be reported as the best concept.

```
Function AdjustThld(queue, $\theta_0$)
    queue $\leftarrow \{c \in$ queue $\mid$ best.IsPromissing($c$)$\}$;
    if queue.Size $< |L|$ then
        return $\theta_0$;
    return $\min \{\theta > \theta_0 \mid L \geq |\{c \in$ queue $\mid \Delta(c) > \theta\}|\}$;
```
Algorithm 3: Adjustment of minimal threshold for Δ-measure.

If yes, in line 8 the quality of *cc* is checked and if it is high the concept is saved as potentially best concept.

In lines 10-11 the concept *cc* is checked. If the value of its Δ-measure is smaller than the threshold θ, it is not Δ-stable concept and, thus, its preimages cannot be Δ-stable either.

Then, in lines 12–13 the concept *cc* is verified if its preimages can potentially be reported as the best concepts, i.e., that the optimimstic estimate for the quality function on its subconcepts (concepts with smaller but comparable extents) is large enough. If yes, the concept *cc* is pushed to the `queue` in line 14 for further processing.

Finally, in lines 15–16 the threshold θ is adjusted in such a way that ensures that the size of the `queue` is smaller than the memory limit L (the parameter of the algorithm). Let us discuss the adjustment in more details.

3.3 Adjustment of Δ threshold

The algorithm for adjusting the threshold is shown in Algorithm 3. First, in line 2 the algorithm removes all patterns that cannot generate concepts with high quality. Then, if necessary it increases the threshold θ in such a way, that the number of concepts with high Δ-measure is less then L.

This adjustment is polynomial, however, the computational complexity of the whole procedure is no more polynomial. Indeed, if the concept with the maximal projection number is expanded, then this procedure become a depth first order FCA algorithm, that requires only $O(|M|)$ memory to store concepts and thus if $L > |M|$ the adjustment procedure is never run and, thus, Algorithm 2 can iterate over all concepts. However, if one always selects the concept with the minimal projection number, then this procedure becomes the same as in Algorithm 1 with the polynomial complexity. Indeed, in this case one can consider the expansions of all concepts with the minimal projection number as one expansion and it become exactly Σοφια algorithm. Similarly, if only expansion of the concepts with small projection number is allowed ,i.e. if the projection number is in the interval $[i_{\min}, i_{\min} + k]$, where $i_{\min}$ is the minimal projection number, gives an intermediate worst-case computational complexity with a multiplier of 2^k. For small k it is small and algorithm can be considered polynomial.

The similar effect on computational complexity can be achieved by always expanding concepts with the largest extent. Such kind of concept expansion corresponds to subgroup discovery procedure. Indeed, the larger the extent, the better the quality Q can be potentially attained on its subsets. Thus, there is a hope, that in practice Algorithm 2 can behave as an algorithm with polynomial computational complexity for subgroup discovery task.

Before proceeding to practical evaluation we should discuss how the best concepts should be registered, i.e. lines 8 and 12 of Algorithm 2.

3.4 Best concept registration

The most simple concept registration procedure can just verify that the quality of the input concept is larger than the quality of the currently known best concept. If the new concept is better, then it substitutes the previous best concept. This procedure has a problem when the Δ-measure threshold θ is adjusted. Indeed, let $\theta = 1$ and the best concept found so far be c_{best}, let $\mathit{\Delta}(c_{best}) = 1$. *If on the next step the stability threshold is adjusted, what should happen to* c_{best}*?*

If it is just preserved, then the result would not correspond to the final projection. Thus, this would invalidate any procedure that compute the statistical significance of the found subgroup [15]. Consequently, a special mechanism is needed allowing for defining the best stable concept found so far for any Δ-threshold ϑ higher than the current threshold. Consequently, any concept that is not dominated either by Δ-measure or by SD quality Q should be registered. Indeed, if for two concepts $\mathit{\Delta}(c_1) \leq \mathit{\Delta}(c_2)$ and $Q(c_1) \leq Q(c_2)$, i.e., c_1 is dominated by c_2, the concept c_1 cannot be the best SD-concept for any θ. By contrast, if $\mathit{\Delta}(c_1) < \mathit{\Delta}(c_2)$ and $Q(c_1) > Q(c_2)$, then for $\theta \leq \mathit{\Delta}(c_1)$ the concept c_1 is Δ-stable and since $Q(c_1) > Q(c_2)$ the concept c_1 is better than c_2 and can be reported as the best concept, however if $\mathit{\Delta}(c_1) < \theta \leq \mathit{\Delta}(c_2)$, then c_1 is no more Δ-stable and thus it cannot be reported as the best concept, but c_2 is still can be reported. Thus, all undominated concepts should be preserved. Thus, Algorithm 4 describes how one should register the best concepts.

In line 2 of Algorithm 4 an element of the `best` concept storage is found. This element is such that $\texttt{best}[i-1] < \mathit{\Delta}(cc) \leq \texttt{best}[i]$ (the set `best` of the best concepts is ordered w.r.t. Δ), i.e., $\texttt{best}[i]$ dominates cc w.r.t. Δ. Thus, if $Q(cc) < Q(\texttt{best}[i])$ the concept cc should not be registered (lines 3–4). Simlarly, in lines 14–15 if the potential of cc is small then it is not promising.

If the concept cc is not dominated by $\texttt{best}[i]$ it is inserted into `best` in lines 5–8. Finally, the quality of cc can be so high that it dominates some concepts from `best` (the concepts $\texttt{best}[j]$ for $j < i$ are already dominated w.r.t. the Δ-measure, thus, they should be compared by means of the quality function). The operations `best.FindInsertPositions`, `best.Insert`, and `best.Remove` are standard operations and can be efficiently implemented by means of red-black trees and other standard approaches.

4 Evaluation

Let us now check how the proposed approach behaves on real data. The approach is evaluated on `chess` dataset from the UCI machine learning repository [9].

```
Function best.Register($cc$)
    $i \leftarrow$ best.FindInsertPosition($\Delta(cc)$);
    if $Q(\texttt{best}[i]) \geq Q(cc)$ then
        return;
    if $\Delta(\texttt{best}[i]) == \Delta(cc)$ then
        best[$i$] $\leftarrow cc$;
    else
        best.Insert($i$,$cc$);
    $i \leftarrow i - 1$;
    while $Q(\texttt{best}[i]) < Q(cc)$ do
        best.Remove($i$);
        $i \leftarrow i - 1$;
Function best.IsPromissing($cc$)
    $i \leftarrow$ best.FindInsertPosition($\Delta_{\mathrm{Proj}(cc)}(cc)$);
    return $Q(\texttt{best}[i]) < \overline{Q}(cc)$;
```

Algorithm 4: Registration of undominated concepts w.r.t. Δ-measure and SD quality Q.

The dataset contains 3196 of objects, corresponding to positions in chess with the supervised information ('the whites win' and 'the whites do not win'). The dataset contains a huge number of concepts and thus it is a good candidate for computational efficiency test. For this task we run the subgroup discovery task for the classification quality function from [13] discussed in Section 1.3.

4.1 Computational time and the dataset size

In the first experiment the size of the dataset is varied from 200 objects to the whole dataset of 3196 objects. For every size a random sample from the original dataset is taken. The memory limit L is also varied from 100 to 100000. The result is shown in Figure 1a. Since it is interesting to see the difference w.r.t. large interval of possible dataset sizes it is given in the log-log scale. However, in order to judge the computational complexity of the approach the real values should be converted to relative values. In particular the time is measured in computational time needed for computing the smallest dataset with the same L. For example, for $L = 1000$ the computation time for the right most point (the whole dataset) is shown to be equal to 2.5, which means that it is in $2^{2.5} = 5.6$ times larger than the time needed for the smallest dataset for the same L. Then the algrithm can be considered linear or better if it is below the function $y = x$.

As it can be seen from the plot for all values of L the computational behaviour of the approach is not worser than linear time complexity. We can also note that the slope of all curves is never larger than 1, which also proves that it behaves linearly w.r.t. the dataset size.

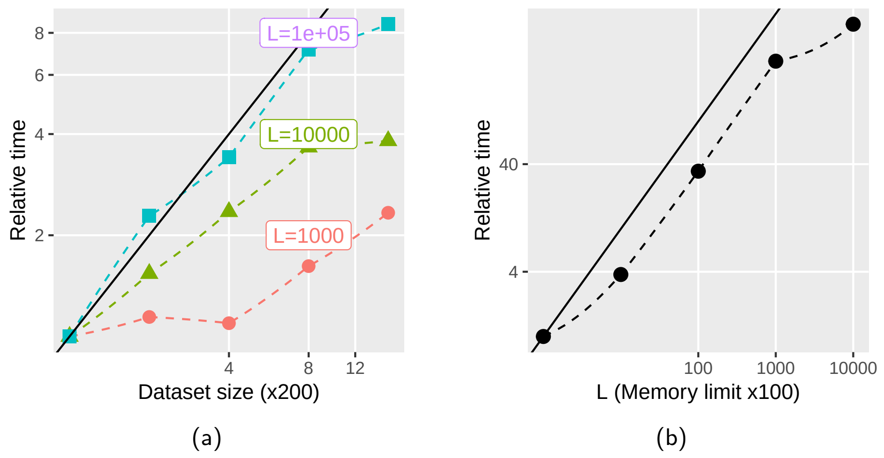

Fig. 1. Computation time w.r.t. dataset size and memory limit

4.2 Computational time and the memory limit

In the second experiment the whole dataset is taken and the memory limit is varying form 100 to 10^6. The result is shown in Figure 1b. As before it is given in the log-log space. As we can see the slope of this curve is also never larger than 1, i.e., the practical computational complexity of this approach is linear.

In both experiments we can see that the slope is much smaller than 1 till a certain moment. It could be explained by the fact that the memory limit is too large and is not totally used. Indeed, when the datasets are small in the first experiments, the total number of concepts is also small, and, thus, the availble memory is too large. However, when the size of the dataset is increased, the total number of concepts is also increased and then the memory limit L become to be important. Similarly, when in the second experiment the memory limit is large ($L = 10^6$), then the memory limit become to be less important.

4.3 Quality of the found concepts

Let us now briefly discuss how the quality of the found concepts depends on different memory limits on the whole dataset. Setting the memory limit to 100, allows finding the best concept with Δ measure of 106, with the extent size of 1002 and the quality of 0.121. By contrast, for $L = 10^6$, the Δ of the found concept is 3, the extent size is 1326 and the quality is 0.157. We see that if we can wait, a better concept can be found, however if the time is limited small L threshold allows finding interesting subgroups much faster.

Conclusion

In this paper a new approach to subgroup discovery was proposed. Although it is not possible to prove polynomial complexity of the algorithm, in practice for subgroup discovery task it shows polynomial behaviour, which is extremly important for processing of large datasets.

References

1. Atzmueller, M.: Subgroup discovery. Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery **5**(1), 35–49 (2015)
2. Belfodil, A., Belfodil, A., Kaytoue, M.: Anytime subgroup discovery in numerical domains with guarantees. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 500–516. Springer (2018)
3. Boley, M., Goldsmith, B.R., Ghiringhelli, L.M., Vreeken, J.: Identifying consistent statements about numerical data with dispersion-corrected subgroup discovery. Data Mining and Knowledge Discovery **31**(5), 1391–1418 (2017)
4. Boley, M., Grosskreutz, H.: Non-redundant Subgroup Discovery Using a Closure System. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 179–194. Springer, Berlin, Heidelberg (2009)
5. Bosc, G., Boulicaut, J.F., Raissi, C., Kaytoue, M.: Anytime discovery of a diverse set of patterns with monte carlo tree search. Data mining and knowledge discovery **32**(3), 604–650 (2018)
6. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Efficient Mining of Subsample-Stable Graph Patterns. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 757–762. New Orlean, LA, USA (2017)
7. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Scalable Estimates of Concept Stability. In: Sacarea, C., Glodeanu, C.V., Kaytoue, M. (eds.) Formal Concept Analysis, LNCS, vol. 8478, pp. 161–176. Springer (2014)
8. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Fast Generation of Best Interval Patterns for Nonmonotonic Constraints. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Gama, J., Jorge, A., Soares, C. (eds.) Machine Learning and Knowledge Discovery in Databases, LNCS, vol. 9285, pp. 157–172. Springer (2015)
9. Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences (2010)
10. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) Conceptual Structures: Broadening the Base, LNCS, vol. 2120, pp. 129–142. Springer (2001)
11. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, 1st edn. (1999)
12. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence **49**(1-4), 101–115 (2007)
13. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup Discovery with CN2-SD. The Journal of Machine Learning Research **5**, 153–188 (2004)
14. Li, G., Zaki, M.J.: Sampling frequent and minimal boolean patterns: theory and application in classification. Data mining and knowledge discovery **30**(1), 181–225 (2016)
15. Pellegrina, L., Riondato, M., Vandin, F.: SPuManTE: Significant Pattern Mining with Unconditional Testing. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining-KDD. vol. 19 (2019)