Pattern Structures for Treatment Optimization in Subgroups of Patients

Natalia V. Korepanova and Sergei O. Kuznetsov

National Research University Higher School of Economics, Moscow, Russia
korepanova.natalia@gmail.com, skuznetsov@hse.ru

Abstract. A comparison of different treatment strategies does not always result in determining the best one for all patients, one needs to study subgroups of patients with significant difference in efficiency between treatment strategies. To solve this problem an approach to subgroups generation is proposed, where data are described in terms of a pattern structure and pattern concepts stay for patient subgroups and their descriptions. To find the most promising pattern concepts in terms of the difference of treatment strategies in efficiency a version of CbO algorithm is proposed. An application to the analysis of data on childhood acute lymphoblastic leukemia is considered.

Keywords: pattern structure, subgroup analysis, acute lymphoblastic leukaemia

1 Introduction

Randomized controlled trial (RCT)[1] is a common approach in evidence-based medicine to prove the superiority of a disease treatment over another one. There are three main types of hypotheses which can be tested with the help of RCT: superiority, noninferiority or equivalence. The main goal of all intervention, drug, or therapy inventions is to find a better way of treating patients. So, superiority trials are possibly the most popular ones, and they are the subject of interest of this paper. A superiority trial allows physicians to find the optimal treatment and improve the curability of the disease. However if comparing treatment strategies are similar enough it is a big success to find and prove the superiority of any of them for all patients. But the effect of treatment strategies may depend on patients initial features (physiological characteristics and/or results of diagnostics). For example, if we compare dosages of a toxic drug a small dosage may be more suitable for patients with light disease manifestation because it throws off a disease and reduces the negative consequences of toxicity while for patients with intense disease manifestation a small dosage is not enough to cure them of the disease. The question is how to find such subgroups of quite similar patients where the efficiency difference of treatment strategies is significant.

Several approaches were proposed in [2–11]. Most of them [5–11] are based on the idea of decision or regression trees which locally optimize some measure at every iteration of the algorithm. This approach is more suitable when we operate

on the big datasets or constantly increasing dataset because they allow to find some subgroups quickly. In the case of treatment optimization the datasets, as a rule, are not very big and do not increase in size rapidly. Moreover, collecting such datasets demands a lot of time and efforts. So, it is more important to carry out more detailed analysis of the data then to make it fast. Also, in RCT on cancer, heart conditions or chronic diseases the outcome of the therapy can be censored, while only few papers like [9, 11] report on analysis of censored data. In this paper we propose a universal approach to finding subgroups of patients with significantly different responses to different treatment strategies, which is not biased by any local optimization criterion. Within this approach subgroups of patients are generated, which are determined by subsets of patients' features.The approach in based on computing closed patterns [14–17] that satisfy criteria of treatment efficiency. The approach was proposed for the analysis of the database of randomized controlled trial on childhood acute lymphoblastic leukemia (ALL) [12, 13] which was performed in several hospitals in Russia and Byelorussia. In this dataset each patient under study is assigned one of the studied treatment strategies, he/she is described by a set of initial features that can be nominal or numerical, and some outcome which is used to estimate treatment efficiency.

The rest of the paper is organized as follows. In section 2 we recall basic definitions of pattern structures and give examples of pattern structures [14–17] relevant to the analyzed data. In section 3 a version of Close-by-One (CbO) algorithm [18] performing on the attributes is proposed. Section 4 presents stopping criterion added to the version of CbO to generate only subgroups with the difference in the efficiency of treatment strategies. Section 5 presents an application of the proposed approach to the ALL dataset, and we conclude in section 6.

2 Pattern Structures

2.1 Main Definitions

In this section we recall pattern structures and examples of pattern structures used for nominal and numerical features.

Let G be a set (of objects), $(D, \sqcap)$ be a meet-semilattice (of all possible object descriptions), and $\delta : G \to D$ be a mapping. Then $(G, (D, \sqcap), \delta)$ is called a *pattern structure*, provided that the set $\delta(G) = \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$, i.e. every subset X of $\delta(G)$ has an infimum $\bigsqcap X$ in $(D, \sqcap)$. Elements of D are called *patterns* and are naturally ordered by subsumption relation $\sqsubseteq$: $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$, where $c, d \in D$. Operation $\sqcap$ is also called a *similarity operation*. If $(G, (D, \sqcap), \delta)$ is a pattern structure we define the derivation operators which form a Galois connection between the powerset of G and $(D, \sqcap)$ as:

$$\begin{array}{ll} A^\diamond = \bigsqcap_{g \in A} \delta(g) & \text{for } A \subseteq G \\ d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} & \text{for } d \in D \end{array} \tag{1}$$

The pairs (A, d) satisfying $A \subseteq G$, $d \in D$, $A^\diamond = d$, and $A = d^\diamond$ are called *pattern concepts* of $(G, (D, \sqcap), \delta)$, with *pattern extent* A and *pattern intent* d. Pattern

concepts are ordered with respect to set inclusion on extents. The ordered set of pattern concepts makes a lattice, called *pattern concept lattice*. Operator $(\cdot)^{\diamond\diamond}$ is an algebraical closure operator on patterns, since it is idempotent, extensive, and monotone.

If objects are described by binary attributes from set M, then $D = \wp(M)$, the powerset of M, and $\delta(g)$ is prime operator $(\cdot)'$ in the context (G, M, I): $\delta(g) = \{m \in M \mid gIm\}$, and $d_1 \sqcap d_2 = d_1 \cap d_2$ where $d_1, d_2 \in D$. So, subsumption corresponds to set inclusion: $d_1 \sqsubseteq d_2 \Leftrightarrow d_1 \sqcap d_2 = d_1 \Leftrightarrow d_1 \cap d_2 = d_1 \Leftrightarrow d_1 \subseteq d_2$.

So, if all patients' initial features are binary we can use them as binary attributes directly. However we also aim at dealing with nominal initial features. Consider patients are described by k initial features $\{\psi_1, ..., \psi_k\}$, all of them are nominal (binary is a particular case) and their values are coded as natural numbers. So, if ψ_i takes l_i values we assume that the range of ψ_i is $\{1, ..., l_i\}$. For each ψ_i we construct l_i binary attributes $\{\beta_i^1, ..., \beta_i^{l_i}\}$ such that $\beta_i^j : G \to \{0, 1\}$ and $\beta_i^j(g) : g \to \psi_i(g) = j$ where $g \in G, j = 1, ..., l_i$. As a result we get $\sum_{i=1,...,n} l_i$ binary attributes to which pattern structures can be applied as it is shown above.

2.2 Pattern Structures on Intervals

To operate with numerical features *interval pattern structures* [15–17] can be applied. Let us consider each patient is described by n numerical and no nominal initial features. In our notation G corresponds to the set of patients. So, let $\{\varphi_1, \varphi_2, ..., \varphi_n\}$ be a set of functions represented patients' initial features such that $\varphi_i : G \to \mathbb{R}$ for $i = 1, ..., n$. For each feature φ_i we construct a corresponding interval attribute $\alpha_i : G \to [\mathbb{R}, \mathbb{R}]$ such that if $\varphi_i(g) = x$ for $g \in G$, then $\alpha_i(g) = [x, x]$, where $x \in \mathbb{R}$. These attributes are used for pattern structures construction.

Each object is described by a n-dimensional tuple of intervals. Let a and b be tuples of n intervals, so $a = \langle [v_i, w_i] \rangle_{i=1,...,n}$ and $b = \langle [x_i, y_i] \rangle_{i=1,...,n}$, where $v_i, w_i, x_i, y_i \in \mathbb{R}\ \forall i = 1, ..., n$. In this case the similarity operation $\sqcap$ is defined by the meet of tuple components:

$$a \sqcap b = \langle [v_i, w_i] \rangle_{i=1,...,n} \sqcap \langle [x_i, y_i] \rangle_{i=1,...,n} = \langle [v_i, w_i] \sqcap [x_i, y_i] \rangle_{i=1,...,n}, \quad (2)$$

where $[v_i, w_i] \sqcap [x_i, y_i] = [\min(v_i, x_i), \max(w_i, y_i)]$.

Hence, subsumption on tuples of interval is defined as:

$$\begin{array}{l} a \sqsubseteq b \Leftrightarrow [v_i, w_i] \sqsubseteq [x_i, y_i]_{i=1,...,n} \Leftrightarrow [v_i, w_i] \sqcap [x_i, y_i] = [v_i, w_i]_{i=1,...,n} \Leftrightarrow \\ \Leftrightarrow [\min(v_i, x_i), \max(w_i, y_i)] = [v_i, w_i]_{i=1,...,n} \Leftrightarrow [v_i, w_i] \sqsupseteq [x_i, y_i]_{i=1,...,n}. \end{array} \quad (3)$$

For example, $\langle [2, 6], [4, 5] \rangle \sqsubseteq \langle [3, 4], [5, 5] \rangle$ as $[2, 6] \sqsubseteq [3, 4]$ and $[4, 5] \sqsubseteq [5, 5]$.

2.3 Pattern Structures on Mixed Tuples

In the previous sections we consider separately the cases of nominal and numerical initial features. However, the situation when patients have both nominal

and numerical initial features seems more natural. As it is described above we associate the set of binary attributes with each nominal feature and an interval attribute - with each numerical one. So, $d \in D$ is a tuple, where components are intervals and binary attributes. Let us have k binary and n interval attributes. Assume $d = \langle \alpha, \beta \rangle$ where α is the tuple of intervals of length n, and β is the subset of binary attributes. If $d_1, d_2 \in D$, $d_1 = \langle \alpha_1, \beta_1 \rangle$, and $d_2 = \langle \alpha_2, \beta_2 \rangle$ similarity operator can be set as $d_1 \sqcap d_2 = \langle \alpha_1 \sqcap \alpha_2, \beta_1 \sqcap \beta_2 \rangle$ where similarity operators for the tuples of intervals and the sets of binary attributes are defined above. The subsumption is also defined by subsumption on the tuples of intervals and the sets of binary attributes:

$$\begin{gathered} d_1 \sqsubseteq d_2 \Longleftrightarrow d_1 \sqcap d_2 = d_1 \Longleftrightarrow \langle \alpha_1, \beta_1 \rangle \sqcap \langle \alpha_2, \beta_2 \rangle = \langle \alpha_1, \beta_1 \rangle \Longleftrightarrow \\ \Longleftrightarrow \alpha_1 \sqcap \alpha_2 = \alpha_1, \beta_1 \sqcap \beta_2 = \beta_1 \Longleftrightarrow \alpha_1 \sqsubseteq \alpha_2, \beta_1 \sqsubseteq \beta_2. \end{gathered} \tag{4}$$

3 Pattern Concepts Generation

Pattern concepts can be computed by Close by One (CbO) algorithm. It produces a tree structure on pattern concepts where edges represent a subset of lattice edges. For the following analysis we do not need the whole lattice but the tree-structure provided by CbO is helpful for implementation of stopping criterion. Considering the top of the lattice is the pair $(G, G^\diamond)$, and the bottom is $(\emptyset, \emptyset^\diamond)$ CbO starts from the bottom and proceeds "object-wise". However, when the number of objects is larger than the number of attributes processing top-down allows one to reduce computation time. Moreover, for the given problem we need to generate pattern concepts with as large extents as possible to detect the difference in treatment efficiency. So, it is more reasonable to start generation process from the top of the lattice. As we operate on descriptions consisting from interval and binary attributes classical CbO must be adapted to such descriptions. For this purpose a version of CbO is proposed below. The idea is to order elements of descriptions, and start to reduce descriptions by changing its elements in this order.

Let objects be described by n interval and k binary attributes. So, we can rewrite description as $d = \langle v_1, w_1, ..., v_n, w_n, b_1, ..., b_k \rangle$, where v_i and w_i are the left and right bounds of the i-th interval attribute for $i = 1, ..., n$, and b_j indicates whether description d contains the j-th binary attribute for $j = 1, ..., k$. So, if b_j is 0 $\delta(g)$ does not contain the j-th binary attribute for all $g \in d^\diamond$, and if b_j is 1, then $\delta(g)$ may or may not contain the j-th binary attribute for all $g \in d^\diamond$. In other words, $1 \sqcap 0 = 0$ or $0 \sqsubseteq 1$. The definition of similarity operator remains the same: we take minimum of left bounds, maximum of right bounds, and set intersection, which in the given notation can be written as element-wise conjunction of indicator vectors:

$$d_1 \sqcap d_2 = \langle \langle \min(v_{1,i}, v_{2,i}), \max(w_{1,i}, w_{2,i}) \rangle_{i=1,...,n}, \langle b_{1,j} \wedge b_{2,j} \rangle_{j=1,...,k} \rangle . \tag{5}$$

The introduced version of CbO starts from the most general description and specifies it by reducing intervals and adding binary features. For interval

reduction it is necessary to choose some step value s_i for each interval attribute ($i = 1, ...n,$). If we aim at some sort of scaling we can set these values by ourselves, or if scaling is unwanted s_i is set to the smallest difference between values of the initial feature corresponding to the i-th interval attribute.

Further we denote $d_{all} = \bigsqcap_{g \in G} \delta(g)$, $min(d_1, d_2)$ denotes the minimum position of unequal elements of tuples d_1 and d_2 in element-wise comparison. Let $suc(d)$ denote the set of all children of the node corresponding to the description d. Let $prev(d)$ return the parent of d, $address(d)$ return the address of d, and $nexti(d)$ store the position of the description tuple d which must be changed at the next algorithm returning to d. Let $\div$ denote integer division, $\%$ denote residue of division, and $[\cdot]$ be an operator of taking the element of the tuple. Function AddConcept set required links between tree nodes when a new node is added. Function OneIteration changes the description d_{curr} given as argument in position $nexti(d_{curr})$ and takes closure of the changed description by $(\cdot)^{\diamond\diamond}$. If min of the closure and d_{curr} is not less than $nexti(d_{curr})$ then the function returns the closure, otherwise it returns d_{curr}.

def AddConcept($parent, child$)
1. $suc(parent) \leftarrow address(child)$
2. $prev(child) := parent$
3. $nexti(child) := nexti(parent)$

def OneIteration(d_{curr})
1. $d_{new} := d_{curr}$
2. $i := nexti(d_{curr})$
3. **if** $i \leq 2n$ **then**
4. $q = i \div 2$
5. $r = i\%2$
6. $d_{new}[i] := d_{new}[i] - s_q(2r - 1)$
7. $d_{add} := d_{new}{}^{\diamond\diamond}$
8. **if** $d_{new}[2q] \leq d_{new}[2q + 1]$ **and** $min(d_{curr}, d_{add}) \geq i$ **then**
9. AddConcept(d_{curr}, d_{add})
10. **return** d_{add}
11. **else return** d_{curr}
12. **else**
13. $d_{new}[i] := d[i] \wedge 1$
14. $d_{add} := d_{new}{}^{\diamond\diamond}$
15. **if** $d_{new}[i] \neq d[i]$ **and** $min(d, d_{add}) \geq i$ **then**
16. AddConcept(d_{curr}, d_{add})
17. **return** d_{add}
18. **else return** d_{curr}

0. $d := d_{all}, nexti(d) := 1, prev(d) := \emptyset, suc(d) := \emptyset$
1. **until** $d = d_{all}$ and $nexti(d) > 2n + k$ **do**
2. **until** $nexti(d) > 2n + k$ **do**
3. d_{add}:=OneIteration(d)
4. **if** $d \neq d_{add}$ **then**

5. $d := d_{add}$
6. $output(d, d^{\diamond})$
7. **else** $nexti(d) := nexti(d) + 1$
8. $d := prev(d)$

Lines 7 and 14 of function OneIteration has complexity $O((2n + k)|G|)$, and all lines 3–8 of the main part of the algorithm are performed at most in this time. The loop starting at line 2 is repeated $2n|G| + k$ times at worst (as in the worst case each boundary of all interval attributes can take $|G|$ values), while the loop starting at line 1 is repeated $|L|$ times exactly, where $|L|$ is the number of pattern concepts. So, the algorithm has complexity $O((2n+k)|G|(2n|G|+k)|L|)$. The complexity is higher than that of CbO, $O((2n + k)|G|^2|L|)$, but in practice our algorithm may become faster when n and k are small, and the number of numerical values in data is less than $|G|$.

4 Stopping Criterion

As it is mentioned above it may not be required to generate all pattern concepts, and the version of CbO may stop when subgroups with difference in treatment efficiency and maximal possible extent are generated. To estimate the difference in efficiency for some description d we define a difference measure which depends on the sets of outcomes of patients who match d and have received the same treatment, and if the value of this measure satisfies some criterion of significance the proposed version of CbO stops to generate specification of the description which is currently in work.

Let p be the number of comparing treatment strategies, and d is the description currently processed by the algorithm. Assume $Q_i = \{outcome(g) \mid g \in d^{\diamond}, treatment(g) = i\}$ for all $i = 1, ..., p$, where $outcome(g)$ is the outcome of g, and $treatment(g)$ is the treatment assigned to g. So, $|Q_i|$ denotes the number of patients received treatment i. We define difference measure μ which takes the sets of outcomes corresponding to each treatment strategy and returns the estimation of difference, and set threshold ε.The criterion looks like if $\mu(Q_1, ..., Q_p) > \varepsilon$ the algorithm does not generate the children of the currently processed node and returns to its parent node.

Except the stopping criterion itself several additional restrictions are necessary. So, if a subgroup does not contain patients per each of p treatment strategies we are not able to compute μ and need to return to the parent subgroup without generating children nodes, since the antimonotonicity of operator $(\cdot)^{\diamond}$ ensures this property for all descendants of the current node. Also, it may be important to result in descriptions with approximately equal number of patients per treatment strategy in corresponding subgroups. Therefore, additional parameter $\lambda \in [0, 1]$ is provided to control the ratio of each treatment strategy in a subgroup (see line 4 in function Restrictions). If λ is set to one this restriction is deactivated, if it is set to zero the numbers of patients per each treatment strategy must be equal. Let $outc(d)$ be $\langle Q_i \rangle_{i=1,...,n}$, $outc(d)[i]$ be Q_i, and $|\cdot|$

return the power of the set. Function Restrictions checks fulfillment of these restrictions for a particular description. Let $isempty(d)$ be $False$ if the subgroups corresponding to description d do not contain patients from every treatment strategy, and $True$ otherwise. Let also $notBalanced(d)$ be $False$ if the subgroup corresponding to d is not balanced (do not satisfy restriction on proportion of treatment strategies in the subgroup), and $True$ otherwise.

def Restrictions(d)
0. $isempty(d) := False, notBalanced(d) := False$
1. **for** j **from** 1 **to** p **do**
2. $\quad isempty(d) := isempty(d) \vee (|outc(d)[j]| = 0)$
3. $\quad share := \frac{|outc(d)[j]|}{|d^\diamond|}$
4. $\quad notBalanced(d) := notBalanced(d) \vee (\frac{1-\lambda}{p} \geq share) \vee (share \geq \frac{1+\lambda}{p})$

0. $d := d_{all}, nexti(d) := 1, prev(d) := \emptyset, suc(d) := \emptyset$
1. Restrictions(d)
2. **until** $d = d_{all}$ **and** $nexti(d) > 2n + k$ **do**
3. $\quad$ **if** $isempty(d)$ **then**
4. $\quad\quad d := prev(d)$
5. $\quad\quad$ **continue**
6. $\quad$ **if** $notBalanced(d)$ **or** $\mu(outc(d)) \leq \varepsilon$ **then**
7. $\quad\quad d_{add}$:=OneIteration(d)
8. $\quad\quad$ **if** $d \neq d_{add}$ **then**
9. $\quad\quad\quad d := d_{add}$
10. $\quad\quad\quad$ Restrictions(d)
11. $\quad\quad$ **else** $nexti(d) := nexti(d) + 1$
12. $\quad$ **else**
13. $\quad\quad output(d, d^\diamond)$
14. $\quad\quad d := prev(d)$

The algorithm outputs the set of maximal size subgroups (i.e. maximal extents) with significant difference in treatment efficiency. Since we do not construct the whole pattern lattice the resulting set may contain subgroups which subsumed under the other subgroups from this set. Smaller subgroups should be excluded from the output by post-processing.

5 Application to ALL Dataset

5.1 Dataset

The dataset consists of more than 2000 patients from 1 to 18 years old with newly diagnosed ALL. All of them were included into the standard risk group (SRG) or into the intermediate risk group (ImRG) of randomized clinical trial MB-ALL-2008 [19]. The protocol of this trial contains three stages of treatment for SRG and ImRG: induction (36 days), consolidation (25 weeks), and maintenance (2-3 years). In this paper we only focus on the induction stage. Induction therapy

aims at bringing a patient into remission and is very toxic. At this stage of treatment patients from SRG and ImRG are randomized into 3 and 2 treatment strategies correspondingly. Let us code them as T_1, T_2, and T_3 in SRG and T_4 and T_5 in ImRG. SRG and ImRG parts of the dataset may be considered as two independent datasets because SRG and ImRG therapies differ considerably.

Each patient from the dataset is described by the set of initial features, treatment strategy which he or she was assigned at randomization, and outcome features. From all initial features 8 were chosen for the analysis: sex (male or female), age (in years from the birth date to the start of the therapy), initial white blood count (per nl) (WBC), immunophenotype (B- or T-ALL), central nervous system (CNS) status (normal, cytosis is less than 5 per mcl and blast cells, neuroleukemia), liver enlargement (in cm), spleen enlargement (in cm), mediastinum status (normal or pathological). As for outcome features of the dataset each outcome feature consists of two parts: the result of the therapy and time from the beginning of the therapy to the date when the result was fixed (in years). In this dataset we have two such features. One of them is fixing the time before patient's death.This feature has three possible states: alive, lost to follow-up or death. When a patient is alive or lost to follow-up we say that censoring happened because we cannot measure exactly the time before death. The other outcome feature represents the time before a negative event . This feature can possess the following states: alive in remission, lost to follow-up, death in remission, secondary tumor, relapse or metastases, nonresponse to the therapy or disease progression, or death in induction. Alive in remission and lost to follow-up events correspond to censoring. Two types of outcomes are used for different variants of treatment efficiency estimation which are presented below.

Finally, we exclude patients with missed values from the further analysis and result in 1221 SRG patients: 387, 366, and 368 patients received T_1, T_2, and T_3 respectively. 929 patients in ImRG: 467 and 462 patients received T_4 and T_5 respectively.

5.2 Data Preprocessing

As it was shown above the initial features of the patients should be transformed into the tuple of interval and binary attributes. Age, WBC, liver enlargement, and spleen enlargement are numerical features, so for each of them an interval attribute is created. Moreover, we scale them a little to obtain subgroup descriptions which make sense for physicians. For example, it is not correct enough in medical terms that one treatment is more effective for patients, let's say, up to 5.4 years old. Similar limitations should also be applied to other 3 numerical features. To answer these limitations we propose the following way of interval attributes construction:

1. Let one set the steps of interval reduction in CbO to $s_{age} = s_{liver} = s_{spleen} = 1$ and $s_{WBC} = 10$.
2. For each $name \in \{age, liver, spleen, WBC\}$ and for every $g \in D$, where D is the set of all patients, if $name(g) = x$ then the value of the corresponding interval attribute is set to $[s_{name} \cdot \lfloor x/s_{name} \rfloor, s_{name} \cdot \lceil x/s_{name} \rceil]$.

All nominal features (sex, immunophenotype, CNS status, and mediastinum status) are converted to the set of binary attributes in the way presented in section 2.1. For instance, we construct two binary attributes corresponding to sex: one indicating males and another indicating females.

5.3 Methodology of Generating Hypotheses

The algorithm of subgroup descriptions generation is proposed in Section 4. It requires to set the difference measure. For the data of the childhood ALL we choose log-rank statistics [20]. As it is a statistical method of detecting the difference between two (or several) survival curves [20–22] the threshold is naturally chosen to satisfy 95% level of confidence of log-rank test. So, if p-value of two-sided log-rank test is less than 5%, we output current description as a potential subgroup description and do not generate its children, otherwise we continue to generate children descriptions in accordance with the introduced algorithm. Finally, from the set of potential subgroup descriptions we delete descriptions subsumed under other potential subgroup descriptions.

As in SRG three treatment strategies (T_1, T_2, T_3) are compared, the algorithm selected those subgroups where the difference between any pair of the treatment strategies is significant. Assume one of descriptions we get is d. For the subgroup described by this description we should compare every pair of treatment strategies: T_1 and T_2, T_1 and T_3, and T_2 and T_3. For each pair where log-rank test detects the difference with confidence 95% and power 80% we make a hypothesis. For instance, if pair T_i and T_j satisfies these requirements then a hypothesis says that treatment strategies T_i and T_j affect patients described by d differently. At the same time if the survival curve for T_i, for instance, is located above the survival curve for T_j we can even say that T_i is better for patients described by d than T_j. For patients from ImRG only two treatment strategies are compared. Therefore it is enough to estimate power, and if it is more than 80% we make a hypothesis in the same way.

5.4 Summary of Results

The proposed algorithm was run to compare separately overall survival (OS)[23], event-free survival (EFS)[24], and relapse-free survival (RFS)[25] in each risk group with λ set to $\frac{1}{3}$ and $\frac{1}{5}$ for SRG and ImRG respectively. We also add restrictions on the size of the subgroups: not less than 20 and 200 patients per each treatment strategy for SRG and ImRG, respectively (the choice is explained by the greater number of patients per treatment and possible descriptions for ImRG). So, as a result three sets of subgroup descriptions were obtained for each risk group (SRG and ImRG), one per each type of survival.

The results of experiments for SRG are presented in Table 1. OS, EFS, and RFS stand for three types of survival described above, CbO stands for the proposed approach, and IT stands for Interaction Tree from [9]. To construct an interaction tree the same restriction on the size of subgroups (i.e. leaves) was set: not less than 20 patients per each treatment strategy. Performing IT with

pruning results in no subgroups, therefore we compared to unpruned trees. To estimate subgroups in each of them paired logrank test p-values for every pair of compared treatment strategies are estimated. We have also performed bootstrap sampling on 1000 samples, and for each subgroup we estimate p-value median and 0.95 unpivotal confidence interval for the difference between long-term survival estimations for every pair of compared treatment strategies. We count the number of subgroups where the result of comparison of even one pair of compared treatment strategies in this subgroups satisfies restrictions at the heading. So, p corresponds to p-value of paired logrank test, p_m is a median estimated by bootstrap, d_l and d_r are the left and the right boundaries of 0.95 bootstrap confidence interval for the difference in long-term survival.

Table 1. Number of subgroups obtained for SRG corresponding to different types of survival and applied apprioach to subgroup detection.

survival type	censoring rate	approach	number of subgroups	$p < 0.05$	$p_m < 0.05$	$p < 0.05$ & $p_m < 0.05$	$p < 0.05$ & $d_l \cdot d_r > 0$	$p < 0.05$ & $p_m < 0.05$ & $d_l \cdot d_r > 0$
OS	0.945	CbO	67	55	50	49	50	46
		IT	10	1	1	1	1	1
EFS	0.923	CbO	166	101	104	100	87	87
		IT	12	1	0	0	0	0
RFS	0.963	CbO	89	54	47	45	21	20
		IT	11	1	1	1	1	1

For ImRG we obtained 2559 and 153 subgroups based on OS and EFS respectively and no subgroups based on RFS since the superiority of T_5 over T_4 in RFS holds for the whole set of ImRG patients. Moreover, all obtained subgroups based on OS and EFS confirm that T_5 is better than T_4. For this reason we did not carry out an experiments on ImRG by applying Interaction Trees.

5.5 An Example of Generated Hypotheses

Given all hypotheses for ImRG propose the superiority of T_5 over T_4 it is more interesting to look at the hypotheses for SRG. Short-term estimation of the treatment strategy efficiencies carried out by physicians shows that T_1 is significantly worse than T_2 and T_3 for all patients from SRG while long-term estimations show no significant difference. However, by applying the proposed algorithm we found several subgroups where T_1 is better than T_2 and T_3 in long-term. For instance, the description was obtained on the basis of difference in OS: $4 \leq$ *age* , $3 \leq$ *liver enlargement* ≤ 7*, and normal mediastinum status.* Testing T_1 vs T_2 and T_1 vs T_3 we got p-values 1.4% and 1.9% and power estimations 86% and 94%. There are approximately 60 patients per treatment strategy in the subgroup. OS curves for the patients which can be described by even one of these descriptions is presented in Fig. 1. Confidence intervals of p-values obtained from 1000 sample bootstrap: [0.3%, 1.6%] and [0.7%, 1.7%]. So, the advantage of strategy

T_1 over T_2 and T_3 for patients matching the description seems confident and independent from the certain data.

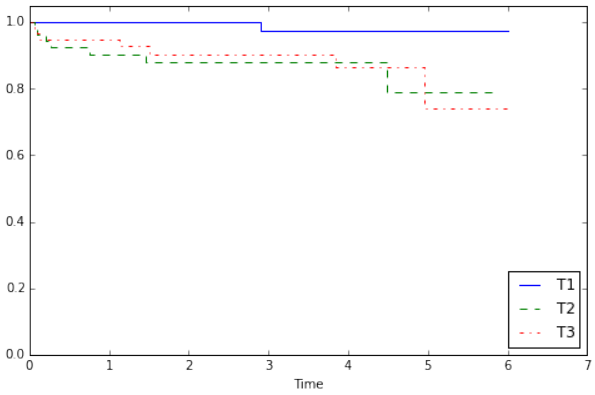

Fig. 1. OS curves for subgroups showing the superiority of T_1 over T_2 and T_3.

6 Conclusion

In this paper we have introduced an approach to solving the problem of determining relevant subgroups of patients for therapy optimization. The approach is based on representing data by numerical pattern structures and applying the version of CbO algorithm. The algorithm computes the pattern lattice top-down (starting with the most general descriptions) and its stopping criterion allows one to generate subgroups with significant differences in the efficiency of treatment strategies containing the maximal possible number of patients to satisfy statistical power restrictions. This approach allows one to avoid binarization or using similarity measures on patients, which can result in artifacts. The approach is also not biased by local optimization heuristics used for constructing decision trees and random forests. The situations when various subgroups are not disjoint will be the subject of further study.

Acknowledgments

We would like to thank Prof. Alexander I. Karachunskiy and his colleagues from the Federal Research Institute of Pediatric Hematology, Oncology and Immunology, Moscow, Russia for helpful discussions and for providing us with data on MB-ALL-2008 protocol.

References

1. Levin, K.A.: Study Design VII. Randomised Controlled Trials. Evident-Based Dentistry 8, 22-23, Nature Publishing Group (2007)
2. Zeileis, A., Hothorn, T., Hornik, K. Model-Based Recursive Partitioning. J. Comput. Graph. Stat. 17(2), 492-514 (2008)
3. Foster, J., Taylor, J., Ruberg, S. Subgroup Identification from Randomized Clinical Trial Data. Stat. Med. 30(24), 2867-2880 (2011)
4. Zhao, Y., Zeng, D., Rush A.J., Kosorok, M.R. Estimating Individualized Treatment Rules Using Outcome Weighted Learning. J. Am. Stat. Assoc. 107(449), 1106-1118 (2012)
5. Korepanova, N., Kuznetsov, S.O., Karachunskiy, A.I.: Matchings and Decision Trees for Determining Optimal Therapy. In: Ignatov D.I. et al. Analysis of Images, Social Networks and Texts, Proc. 3rd International Conference (AIST2014), Communications in Computer and Information Science, Vol. 436, pp. 101-110. Springer (2014)
6. Dusseldorp, E., Conversano ,C., Van Os, B.J. Combining an Additive and Tree-Based Regression Model Simultaneously: STIMA. J. Comput. Graph. Stat. 19(3), 514-530 (2010)
7. Dusseldorp, E., Van Mechelen, I. Qualitative Interaction Trees: a Tool to Identify Qualitative Treatment-Subgroup Interactions. Stat. Med. 33, 219–237 (2014)
8. Lipkovich, I., Dmitrienko, A., Denne, J., Enas, G. Subgroup Identification Based on Differential Effect Search a Recursive Partitioning Method for Establishing Response to Treatment in Patient Subpopulations. Stat. Med. 30(21), 2601-2621 (2011)
9. Su, X., Zhou, T., Yan, X., Fan, J., Yang, S. Interaction Trees with Censored Survival Data. Int. J. Biostat. 4(1), 2 (2008)
10. Su, X., Tsai, C.L., Wang, H., Nickerson, D.M., Li, B. Subgroup Analysis via Recursive Partitioning. J. Mach. Learn. Res. 10, 141-158 (2009)
11. Loh, W.-Y., He, X., Man, M. A Regression Tree Approach to Identifying Subgroups with Differential Treatment Effects. Stst. Med. 34, 1818 – 1833 (2015)
12. Ching-Hon, P., Robison, L.L., Look, A.T. Acute Lymphoblastic Leukaemia. The Lancet. 9617(371), 1030 – 1043, The Lancet (2008)
13. Karachunskiy, A., Roumiantseva, J., Lagoiko, S. (eds.) Efficacy and Toxicity of Dexamethasone vs Methylprednisolone Long-Term Results in More than 1000 Patients from the Russian Randomized Multicentric trial ALL-MB 2002, Letter to the Editor. Leukemia. 29, 1955–1958, Nature Publishing Group (2015)
14. Ganter, B., Kuznetsov, S.O. Pattern Structures and Their Projections. In: Proc. Stumme, G., Delugach, H. (eds.) 9th International Conference on Conceptual Structures (ICCS 2001). Lecture Notes in Artificial Intelligence, vol. 2120, pp. 129–142, Springer (2001)
15. Kuznetsov, S.O. Pattern Structures for Analyzing Complex Data. In: Sakai H. et al. (eds.) Proc. 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2009). Lecture Notes in Artificial Intelligence, vol. 5908, pp. 33–44, Springer (2009)
16. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S. Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis. Information Sciences, 10(181), 1989–2001, Elsevier, New York (2011)
17. Kuznetsov, S.O. Scalable Knowledge Discovery in Complex Data with Pattern Structures. In: Maji, P., Ghosh, A., et al. (eds.) Proc. 5th International Conference

Pattern Recognition and Machine Intelligence (PReMI'2013). Lecture Notes in Computer Science, vol. 8251, pp. 30–41, Springer (2013)
18. Kuznetsov, S.O. A Fast Algorithm for Computing All Intersections of Objects from an Arbitrary Semilattice. Nauchno-Tekhnicheskaya Informatsiya Seriya 2 - Informatsionnye Protsessy i Sistemy, 1, 17–20 (1993).
19. A service of the U.S. National Institutes of Health, https://clinicaltrials.gov/ct2/show/NCT01953770
20. Kleinbaum, D.G., Klein, M.: Kaplan-Meier Survival Curves and the Log-Rank Test. In: Survival Analysis, pp. 55–96, Springer New York (2012)
21. Kaplan, E.L., Meier, P. Nonparametric Estimation from Incomplete Observations. Journal of the American Statistical Association, 282(53), 457–481 (1958)
22. May, W.L. Kaplan-Meier Survival Analysis. In: Encyclopedia of Cancer, pp. 1590-1593, Springer Berlin Heidelberg (2009)
23. National Cancer Institute, http://www.cancer.gov/publications/dictionaries/cancer-terms?cdrid=655245
24. National Cancer Institute, http://www.cancer.gov/publications/dictionaries/cancer-terms?cdrid=655269
25. National Cancer Institute, http://www.cancer.gov/publications/dictionaries/cancer-terms?cdrid=655254