Action Rules and the GUHA Method: Preliminary Considerations and Results

Jan Rauch and Milan Šimůnek

Faculty of Informatics and Statistics, University of Economics, Prague*
nám W. Churchilla 4, 130 67 Prague, Czech Republic
rauch@vse.cz, simunek@vse.cz

Abstract. The paper presents an alternative approach to action rules. The presented approach is based on experience with the GUHA method and the LISp-Miner system. G-action rules are introduced. First experience with new GUHA procedure Ac4ft-Miner that mines for G-action rules is described.

1 Introduction

Action rules present a promising direction in data mining research. The notion of action rules was proposed in [4]. The basic idea of action rules is to suggest a way to re-classify objects (for instance customers) to a desired state. The action rules are based on stable and flexible attributes [4]. An example of a *stable attribute* is the date of birth. An example of a *flexible attribute* is an interest rate on a customer account that depends on a bank. Action rules suggest a way how to change the values of flexible attributes to get a desired state. An example is a suggestion to change the interest rate to decrease customer attrition. There are various approaches to mine action rules, see e.g. [4,5,7].

This paper introduces an approach to action rules based on the GUHA method and its implementation in the LISp-Miner system [10,11]. This approach offers new ways how to take an action to get an advantage.

The paper is organized as follows. The action rules are introduced in section 2. Important features of the GUHA method and the LISp-Miner system are summarized in section 3. Definition of action rules based on these features is given in section 4. We call such action rules *G-action rules.* First experience with a new GUHA procedure *Ac4ft-Miner* that mines for G-action rules is presented in section 5. Conclusions and description of further work are in section 6.

2 Action Rules

We start with action rules defined in [5]. There are various additional approaches to action rules, some of them are closely related to the approach introduced in

* The work described here has been supported by Grant No. 201/08/0802 of the Czech Science Foundation and by Grant No. ME913 of Ministry of Education, Youth and Sports, of the Czech Republic.

J. Rauch et al. (Eds.): ISMIS 2009, LNAI 5722, pp. 76–87, 2009.

[5]. An overview paper related to action rules is [6], an example of another paper is [7].

Action rules are in [5] defined on the basis of an information system $\mathcal{S} = (\mathcal{U}, \mathcal{A})$ where $\mathcal{U}$ is a nonempty, finite set of objects and $\mathcal{A}$ is a nonempty, finite set of attributes. It means that each $A \in \mathcal{A}$ is a function $\mathcal{U} \rightarrow V_A$ where V_A is a domain of A. A special type of information system is called *decision table*. It is any information system $\mathcal{S} = (\mathcal{U}, \mathcal{A}_{St} \cup \mathcal{A}_{Fl} \cup \{D\})$ where $D \notin \mathcal{A}_{St} \cup \mathcal{A}_{Fl}$ is a distinguished attribute called a decision and the set $\mathcal{A}$ of attributes is partitioned into stable conditions $\mathcal{A}_{St}$ and flexible conditions $\mathcal{A}_{Fl}$. Action rule R in S is an expression

$$(A_1 = \omega_1) \wedge \ldots \wedge (A_q = \omega_q) \wedge (B_1, \alpha_1 \rightarrow \beta_1) \wedge \ldots \wedge (B_p, \alpha_p \rightarrow \beta_p) \Rightarrow (D, k_1 \rightarrow k_2)$$

where $\{B_1, \ldots, B_p\}$ are flexible attributes and $\{A_1, \ldots, A_q\}$ are stable in $\mathcal{S}$. Moreover, it is assumed that $\omega_i \in Dom(A_i)$, $i = 1, \ldots, q$ and $\alpha_i, \beta_i \in Dom(B_i)$, $i = 1, \ldots, p$. The term $(A_i = \omega_i)$ means that the value of the attribute A_i is ω_i. The term $(B_j, \alpha_j \rightarrow \beta_j)$ means that the value of the attribute B_j has been changed from α_j to β_j, similarly for $(D, k_1 \rightarrow k_2)$.

The left hand side pattern of the above action rule is the set $P_L = V_L \cup \{k_1\}$ where $V_L = \{\omega_1, \ldots, \omega_q, \alpha_1, \ldots, \alpha_p\}$. The domain $Dom_{\mathcal{S}}(V_L)$ of P_L is a set of objects in $\mathcal{S}$ that exactly match V_L. Card$[Dom_{\mathcal{S}}(V_L)]$ is the number of objects in $Dom_{\mathcal{S}}(V_L)$, Card$[Dom_{\mathcal{S}}(P_L)]$ is the number of objects that exactly match P_L, and Card$[\mathcal{U}]$ is the total number of objects in $\mathcal{S}$. The left support $supL(R)$ of the action rule R is defined as $supL(R) = \text{Card}[Dom_{\mathcal{S}}(P_L)]/\text{Card}[\mathcal{U}]$.

The right support $supR(R)$ of the action rule R is defined analogously i.e. $supR(R) = \text{Card}[Dom_{\mathcal{S}}(P_R)]/\text{Card}[\mathcal{U}]$ where $P_R = V_R \cup \{k_2\}$ and V_R is defined as $V_R = \{\omega_1, \ldots, \omega_q, \beta_1, \ldots, \beta_p\}$.

The *support of the action rule* R in $\mathcal{S}$ is denoted by $Sup_{\mathcal{S}}(R)$ and it is the same as the left support $supL(R)$. The *confidence* of the action rule R in $\mathcal{S}$ is denoted by $Conf_{\mathcal{S}}(R)$ and it is defined as

$$(\text{Card}[Dom_{\mathcal{S}}(P_L)]/\text{Card}[Dom_{\mathcal{S}}(V_L)]) * (\text{Card}[Dom_{\mathcal{S}}(P_R)]/\text{Card}[Dom_{\mathcal{S}}(V_R)]) \ .$$

An algorithm for mining of such action rules is described in [5] together with discussion of additional approaches.

3 The GUHA Method and the LISP-Miner System

GUHA is a method of exploratory data analysis developed since 1960's [2]. Its goal is to offer all interesting facts following from the analyzed data to the given problem. GUHA is realized by GUHA-procedures. Input of a GUHA-procedure consists of the analyzed data and of a simple definition of a usually very large set of relevant (i.e. potentially interesting) patterns. The procedure generates each particular pattern and tests if it is true in the analyzed data. The output of the procedure consists of all prime patterns. The pattern is prime if it is true in the analyzed data and if it does not immediately follow from the other more simple output patterns [3].

The most important GUHA procedure is the procedure ASSOC [3]. It mines for patterns that can be understood as association rules, they are however more general than the "classical" association rules defined in [1]. The probably most used implementation of the ASSOC procedure is the procedure *4ft-Miner*. It has various new important features and it mines also for conditional association rules [10].

Implementations of the procedure ASSOC are based on *representation of analyzed data by strings of bits* [8,10], the well known apriori algorithm [1] is not used. A system of modules for dealing with strings of bits was developed [13]. Their utilization leads to algorithm with the complexity linearly dependent on the number of rows of the analyzed data matrices [10]. These modules were used to implement five additional GUHA procedures, all of them are included in the LISp-Miner system [11]. This paper describes a new GUHA procedure *Ac4ft-Miner* that mines for patterns that can be understood as an enhancement of action rules introduced in Section 2.

All the GUHA procedures implemented in the LISp-Miner system deal with data matrices. An example of the data matrix is the data matrix $\mathcal{M}$ shown in Fig. 1.

	stable attributes			flexible attributes				examples of basic Boolean attributes		
object	A_1	$\dots$	A_Q	B_1	$\dots$	B_P	D	$A_1(2)$	$B_1(9,12)$	$D(5,7,9)$
o_1	6	$\dots$	4	12	$\dots$	9	7	0	1	1
o_2	13	$\dots$	2	9	$\dots$	5	3	0	1	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
o_n	2	$\dots$	2	8	$\dots$	5	6	1	0	0

Fig. 1. Data matrix $\mathcal{M}$ with stable and flexible attributes

Rows of a data matrix correspond to observed objects (e.g. clients of a bank), columns correspond to attributes describing properties of particular objects (e.g. date of birth or interest rate on a customer account), possible values of attributes are called categories. Data matrix $\mathcal{M}$ in Fig. 1 has attributes $A_1, \dots, A_P, B_1, \dots, B_Q, D$. Each data matrix corresponds to an information system $\mathcal{S} = (\mathcal{U}, \mathcal{A})$ in a clear way. In the case of data matrix $\mathcal{M}$ from the Fig. 1 it is $\mathcal{U} = \{o_1, \dots, o_n\}$ etc. If we consider also information on stable and flexible attributes as given in first row of Fig. 1 then the data matrix $\mathcal{M}$ can be seen as a decision table $\mathcal{S} = (\mathcal{U}, \mathcal{A}_{St} \cup \mathcal{A}_{Fl} \cup \{D\})$ where $\mathcal{A}_{St} = \{A_1, \dots, A_P\}$ and $\mathcal{A}_{Fl} = \{B_1, \dots, B_Q\}$.

The procedure *Ac4ft-Miner* was derived from the procedure *4ft-Miner* that mines for association rules of the form $\varphi \approx \psi$ where φ and ψ are Boolean attributes. The Boolean attribute φ is called *antecedent* and ψ is called *succedent*. The association rule $\varphi \approx \psi$ means that φ and ψ are associated in the way given by the symbol $\approx$. The symbol $\approx$ is called the *4ft-quantifier*. It corresponds to a condition concerning a four-fold contingency table of φ and ψ. Various types of dependencies of φ and ψ can be expressed by 4ft-quantifiers. The rule $\varphi \approx \psi$ is *true in data matrix* $\mathcal{M}$ if the condition corresponding to the 4ft-quantifier is satisfied in the four-fold contingency table of φ and ψ in $\mathcal{M}$, otherwise $\varphi \approx \psi$ is *false in data matrix* $\mathcal{M}$.

The four-fold contingency table of φ and ψ in data matrix $\mathcal{M}$ is a quadruple $\langle a, b, c, d \rangle$ of natural numbers such that a is the number of rows of $\mathcal{M}$ satisfying both φ and ψ, b is the number of rows of $\mathcal{M}$ satisfying φ and not satisfying ψ, etc., see Table 1. The four-fold contingency table (the *4ft table*) of φ and ψ in $\mathcal{M}$ is denoted by $4ft(\varphi, \psi, \mathcal{M})$.

Table 1. 4ft table $4ft(\varphi, \psi, \mathcal{M})$ of φ and ψ in $\mathcal{M}$

$\mathcal{M}$	ψ	$\neg\psi$
φ	a	b
$\neg\varphi$	c	d

There are 14 basic 4ft-quantifiers implemented in the 4ft-Miner procedure, it is possible to use also conjunctions of basic 4ft-quantifiers. A simple example is the 4ft-quantifier $\Rightarrow_{p,B}$ of *founded implication* [3] that is defined for $0 < p \leq 1$ and $B > 0$ by the condition $\frac{a}{a+b} \geq p \wedge a \geq B$. The association rule $\varphi \Rightarrow_{p,B} \psi$ means that at least $100p$ percent of rows of $\mathcal{M}$ satisfying φ satisfy also ψ and that there are at least B rows of $\mathcal{M}$ satisfying both φ and ψ.

The Boolean attributes φ and ψ are derived from the columns of data matrix $\mathcal{M}$. We assume there is a finite number of possible values for each column of $\mathcal{M}$. *Basic Boolean attributes* are created first. The basic Boolean attribute is an expression of the form $A(\kappa)$ where $\kappa \subset \{a_1, \ldots a_k\}$ and $\{a_1, \ldots a_k\}$ is the set of all possible values of the column A. The basic Boolean attribute $A(\kappa)$ is true in row o of $\mathcal{M}$ if it is $a \in \alpha$ where a is the value of the attribute A in row o. The set κ is called a *coefficient* of basic Boolean attribute $A(\kappa)$. For example {9,12} is a coefficient of the basic Boolean attribute $B_1(9, 12)$. There are examples of values of basic Boolean attributes in Fig. 1; the value 1 means *true* and the value 0 means *false*. Boolean attributes φ and ψ are derived from basic Boolean attributes using propositional connectives $\vee$, $\wedge$ and $\neg$ in the usual way.

The input of the procedure *4ft-Miner* consists of the analyzed data matrix and of several parameters defining (usually very) large set of association rules to be verified. There are very fine tools to define this set [10], some of them are introduced in Sect. 5.

4 G-Action Rules

A G-action rule $\mathcal{R}$ is an expression of the form $\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ where φ_{St} is a Boolean attribute called *stable antecedent*, Φ_{Chg} is an expression called *change of antecedent*, ψ_{St} is a Boolean attribute called *stable succedent*, Ψ_{Chg} is an expression called *change of succedent*, and $\approx^*$ is a symbol called

Ac4ft-quantifier. An example of the Ac4ft-quantifier is the symbol $\Rightarrow_{p,B_1,B_2}$. An example of the G-action rule is the expression

$$A_1(\kappa_1) \wedge A_2(\kappa_2) \wedge [B_1(\lambda_1) \rightarrow B_1(\lambda_1')] \Rightarrow_{p_1 \rightarrow p_2, B_1, B_2} A_3(\kappa_3) \wedge [B_2(\lambda_2) \rightarrow B_2(\lambda_2')] \ .$$

Both the *change of antecedent* Φ_{Chg} and the *change of succedent* Ψ_{Chg} are built from *changes of coefficient*. The *change of coefficient* is an expression $[Z(\kappa) \rightarrow Z(\kappa')]$ where both $Z(\kappa_1)$ and $Z(\kappa_1)$ are literals with coefficients κ and κ' respectively such that $\kappa \cap \kappa' = \emptyset$. The *change of Boolean attribute* is created from *changes of coefficient* and Boolean connectives in the same way as the Boolean attribute is created from the literals, see also Section 3. Both the *change of antecedent* and the *change of succedent* are *changes of Boolean attribute.*

If $\Lambda = [Z(\kappa) \rightarrow Z(\kappa')]$ is a change of coefficient, then an *initial state* $\mathcal{I}(\Lambda)$ of Λ is defined as $\mathcal{I}(\Lambda) = Z(\kappa)$ and a *final state* $\mathcal{F}(\Lambda)$ of Λ is defined as $\mathcal{F}(\Lambda) = Z(\kappa')$. If Φ is a change of Boolean attribute, then *initial state* $\mathcal{I}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals Λ occurring in Φ by their initial states $\mathcal{I}(\Lambda)$. The *final state* $\mathcal{F}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals Λ occurring in Φ by their final states $\mathcal{F}(\Lambda)$. Two examples: $\mathcal{I}(\ A_1(\kappa_1) \wedge A_2(\kappa_2) \wedge [B_1(\lambda_1) \rightarrow B_1(\lambda_1')]\ ) \ = A_1(\kappa_1) \wedge A_2(\kappa_2) \wedge B_1(\lambda_1)$ and $\mathcal{F}(\ A_3(\kappa_3) \wedge [B_2(\lambda_2) \rightarrow B_2(\lambda_2')]\ ) = A_3(\kappa_3) \wedge B_2(\lambda_2')$.

The attributes used in Φ_{Chg} are called *independent attributes* of $\mathcal{R}$ and the attributes used in Ψ_{Chg} are called *dependent attributes* of $\mathcal{R}$. The action rule $\mathcal{R} : \varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ says what happen with objects satisfying stable conditions φ_{St} and ψ_{St} when we change values of their flexible independent attributes in the way given by Φ_{Chg} – i.e. from the initial state characterized by the Boolean attribute $\mathcal{I}(\Phi_{Chg})$ into a final state characterized by the Boolean attribute $\mathcal{F}(\Phi_{Chg})$. The effect is described by two association rules $\mathcal{R}_I$ and $\mathcal{R}_F$:

$$\mathcal{R}_I : \varphi_{St} \wedge \mathcal{I}(\Phi_{Chg}) \approx_I \psi_{St} \wedge \mathcal{I}(\Psi_{Chg}) \ \ \mathcal{R}_F : \varphi_{St} \wedge \mathcal{F}(\Phi_{Chg}) \approx_F \psi_{St} \wedge \mathcal{F}(\Psi_{Chg}).$$

The first rule $\mathcal{R}_I$ characterizes the initial state. The second rule $\mathcal{R}_F$ describes the final state induced by the change of the independent flexible attributes property (i.e. Boolean attribute) $\mathcal{I}(\Phi_{Chg})$ to $\mathcal{F}(\Phi_{Chg})$. If we denote $\varphi_{St} \ \wedge \ \mathcal{I}(\Phi_{Chg})$ as φ_I, $\psi_{St} \ \wedge \ \mathcal{I}(\Psi_{Chg})$ as ψ_I, $\varphi_{St} \ \wedge \ \mathcal{F}(\Phi_{Chg})$ as φ_F, and $\psi_{St} \ \wedge \ \mathcal{F}(\Psi_{Chg})$ as ψ_F then the rules $\mathcal{R}_I$ and $\mathcal{R}_F$ can be written as

$$\mathcal{R}_I : \qquad \varphi_I \approx_I \psi_I \qquad\qquad \mathcal{R}_F : \qquad \varphi_F \approx_F \psi_F \ .$$

The action rule $\mathcal{R}$ makes possible to see the effect of the change Φ_{Chg} in three steps: (1) φ_F is true instead of φ_I. (2) The values of dependent flexible attributes are changed such that $\mathcal{F}(\Psi_{Chg})$ is true instead of $\mathcal{I}(\Psi_{Chg})$ and thus ψ_F is true instead of ψ_I. (3) The initial relation of $\varphi_I \approx_I \psi_I$ described by the 4ft-quantifier $\approx_I$ is changed to the final relation $\varphi_F \approx_F \psi_F$ described by the 4ft-quantifier $\approx_F$.

The truthfulness of the G-action rule $\mathcal{R}$: $\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ is defined on the basis of this interpretation of the change Φ_{Chg}. The G-action rule $\mathcal{R}$ is true in the analyzed data matrix $\mathcal{M}$ if the condition corresponding to the *Ac4ft-quantifier* $\approx^*$ is satisfied in the data matrix $\mathcal{M}$. The sense of the rule $\mathcal{R}$ is expressed by the rules $\mathcal{R}_I$ and $\mathcal{R}_F$ described above. Thus the condition related to $\approx^*$ and defining the truthfulness of $\mathcal{R}$ is related to a way in which the rules $\mathcal{R}_I$ and $\mathcal{R}_F$ are evaluated. The rule $\mathcal{R}_I$ is evaluated on the basis of 4ft-table $4ft(\varphi_I, \psi_I, \mathcal{M})$ of φ_I and ψ_I in $\mathcal{M}$ and the rule $\mathcal{R}_F$ is evaluated on the basis of 4ft-table $4ft(\varphi_F, \psi_F, \mathcal{M})$ of φ_F and ψ_F in $\mathcal{M}$, see Fig. 2.

$\mathcal{M}$	ψ_I	$\neg\psi_I$
φ_I	a_I	b_I
$\neg\varphi_I$	c_I	d_I

4ft-table $4ft(\varphi_I, \psi_I, \mathcal{M})$

$\mathcal{M}$	ψ_F	$\neg\psi_F$
φ_F	a_F	b_F
$\neg\varphi_F$	c_F	d_F

4ft-table $4ft(\varphi_F, \psi_F, \mathcal{M})$

Fig. 2. The 4ft-tables $4ft(\varphi_I, \psi_I, \mathcal{M})$ and $4ft(\varphi_F, \psi_F, \mathcal{M})$

An example of Ac4ft-quantifier $\approx^*$ is the Ac4ft-quantifier $\Longrightarrow^{I>F}_{q,B_1,B_2}$ defined by the condition $\frac{a_I}{a_I+b_I} - \frac{a_F}{a_F+b_F} \geq q \wedge a_I \geq B_1 \wedge a_F \geq B_2$. It is assumed that $0 < q \leq 1$, $B_1 > 0$, and $B_2 > 0$.

If the action rule $\mathcal{R} = \varphi_{St} \wedge \Phi_{Chg} \Rightarrow_{p,B_1,B_2} \psi_{St} \wedge \Psi_{Chg}$ with Ac4ft-quantifier $\Longrightarrow^{I>F}_{q,B_1,B_2}$ is true in the data matrix $\mathcal{M}$ then its effect can be expressed by two association rules $\mathcal{R}_I$ and $\mathcal{R}_F$ with 4ft-quantifiers $\Rightarrow_{=,p+q,B_1}$ and $\Rightarrow_{=,p,B_2}$:

$$\mathcal{R}_I : \quad \varphi_I \Rightarrow_{=,p,B_1} \psi_I \qquad\qquad \mathcal{R}_F : \quad \varphi_F \Rightarrow_{=,p+q,B_2} \psi_F$$

where $\varphi_{St} \wedge \mathcal{I}(\Phi_{Chg})$ is denoted by φ_I, $\psi_{St} \wedge \mathcal{I}(\Psi_{Chg})$ by ψ_I, $\varphi_{St} \wedge \mathcal{F}(\Phi_{Chg})$ by φ_F, and $\psi_{St} \wedge \mathcal{F}(\Psi_{Chg})$ by ψ_F, see also above. The 4ft-quantifier $\Rightarrow_{=,p,B}$ is derived from the 4ft-quantifier $\Rightarrow_{p,B}$, $\Rightarrow_{=,p,B}$ is defined for $0 < p \leq 1$ and $B > 0$ by the condition $\frac{a}{a+b} = p \wedge a = B$, see Tab. 1. Remember that 4ft-quantifier $\Rightarrow_{p,B}$ is defined by the condition $\frac{a}{a+b} \geq p \wedge a \geq B$, see Sect. 3. The parameter p in $\Rightarrow_{=,p+q,B_1}$ and $\Rightarrow_{=,p,B_2}$ depends on the data matrix $\mathcal{M}$.

Informally speaking, the Ac4ft-quantifier $\Longrightarrow^{I>F}_{q,B_1,B_2}$ expresses the fact that the confidence of the rule $\mathcal{R}_F$ is smaller than the confidence of the rule $\mathcal{R}_I$. It is suitable when the truthfulness of the attribute ψ_{St} is undesirable. In the case when the truthfulness of the attribute ψ_{St} is desirable we can use Ac4ft-quantifier $\Longrightarrow^{F>I}_{q,B_1,B_2}$ defined by the condition $\frac{a_F}{a_F+b_F} - \frac{a_I}{a_I+b_I} \geq q \wedge a_I \geq B_1 \wedge a_F \geq B_2$.

We use examples concerning the data set STULONG described at the website `http://euromise.vse.cz/challenge2004/` [1]. Data set consists of four data matrices, we deal with data matrix *Entry* only. It concerns 1 417 patients, each row describes one patient. Data matrix has 64 columns corresponding to particular attributes – characteristics of patients. In a following example we use attributes *Height* (in cm), *BMI* (Body Mass Index), *Cholesterol* (in mg%).

[1] The study (STULONG) was realized at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital (head. Prof. M. Aschermann, MD, SDr, FESC), under the supervision of Prof. F. Boudík, MD, ScD, with collaboration of M. Tomečková, MD, PhD and Ass. Prof. J. Bultas, MD, PhD. The data were transferred to the electronic form by the EuroMISE Centre of Charles University and Academy of Sciences (head. Prof. RNDr. J. Zvárová, DrSc).

An example of the action rule is the rule $\mathcal{R}_1$

$$\mathit{Height}\langle 163, 175) \wedge [\mathit{BMI}(> 30) \rightarrow \mathit{BMI}(27; 30\rangle] \Longrightarrow^{I>F}_{0.119,11,12} \mathit{Cholesterol}(\geq 290)$$

that is described by two association rules $\mathcal{R}_{1I}$ and $\mathcal{R}_{1F}$ with 4ft-quantifiers $\Rightarrow_{=,0.204,11}$ and $\Rightarrow_{=,0.085,12}$:

$$\mathcal{R}_{1I} : \quad \mathit{Height}\langle 163, 175) \wedge \mathit{BMI}(> 30) \Rightarrow_{=,0.204,11} \mathit{Cholesterol}(\geq 290)$$

$$\mathcal{R}_{1F} : \quad \mathit{Height}\langle 163, 175) \wedge \mathit{BMI}(24; 27\rangle \Rightarrow_{=,0.085,20} \mathit{Cholesterol}(\geq 290) \ .$$

The corresponding 4ft-tables $4ft(\mathit{Height} \wedge \mathit{BMI}_I, \mathit{Chlst}, \text{STULONG})$ and $4ft(\mathit{Height} \wedge \mathit{BMI}_F, \mathit{Chlst}, \text{STULONG})$ are in Fig. 3. We denote $\mathit{Height}\langle 163, 175)$ as *Height*, $\mathit{BMI}(> 30)$ as BMI_I, $\mathit{BMI}(24; 27\rangle$ as BMI_F, and $\mathit{Cholesterol}(\geq 290)$ as *Chlst*.

STULONG	*Chlst*	$\neg$*Chlst*
$\mathit{Height} \wedge \mathit{BMI}_I$	11	43
$\neg(\mathit{Height} \wedge \mathit{BMI}_I)$	128	1235

$4ft(\mathit{Height} \wedge \mathit{BMI}_I, \mathit{Chlst}, \text{STULONG})$

STULONG	*Chlst*	$\neg$*Chlst*
$\mathit{Height} \wedge \mathit{BMI}_F$	12	130
$\neg(\mathit{Height} \wedge \mathit{BMI}_F)$	127	1148

$4ft(\mathit{Height} \wedge \mathit{BMI}_F, \mathit{Chlst}, \text{STULONG})$

Fig. 3. 4ft-tables for association rules $\mathcal{R}_{1I}$ and $\mathcal{R}_{1F}$

Height is a stable attribute and *BMI* is a flexible attribute that can be influenced by the patient. The Boolean attribute $\mathit{Cholesterol}(\geq 290)$ means that the level of cholesterol is too high. Attribute *Cholesterol* is here considered as stable even if the level of cholesterol can be also influenced by the patient. The rule $\mathcal{R}_1$ describes how the probability of having $\mathit{Cholesterol}(\geq 290)$ can be influenced by change of *BMI* for patients with height in interval $\langle 163, 175)$. Its message is that among patients satisfying $\mathit{Height}\langle 163, 175) \wedge \mathit{BMI}(> 30)$ are 20,4 percent of patients satisfying $\mathit{Cholesterol}(\geq 290)$ but among patients satisfying $\mathit{Height}\langle 163, 175) \wedge \mathit{BMI}(24; 27\rangle$ only 8,5 percent of patients satisfy $\mathit{Cholesterol}(\geq 290)$.

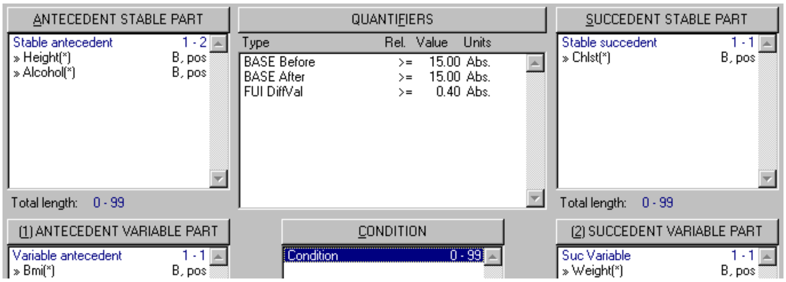
5 Ac4ft-Miner

The *Ac4ft-Miner* is a GUHA procedure. It means that its input consists of a relatively simple definition of a large set $\varOmega$ of relevant G-action rules of the form

$$\varphi_{St} \wedge \varPhi_{Chg} \approx^{*} \psi_{St} \wedge \varPsi_{Chg}$$

and of an analyzed data matrix $\mathcal{M}$. The *Ac4ft-Miner* procedure generates all G-action rules $\omega \in \varOmega$ and verifies each of them in the analyzed data matrix $\mathcal{M}$. The output of the *Ac4ft-Miner* consists of all G-action rules ω true in $\mathcal{M}$.

The set $\varOmega$ is given by definitions of the set $\mathcal{B}_{A,St}$ of relevant Boolean attributes considered as stable antecedents, the set $\mathcal{C}_A$ of relevant changes of antecedent, the set $\mathcal{B}_{S,St}$ of Boolean attributes considered as relevant stable succedents, the set $\mathcal{C}_S$ of relevant changes of succedent, and the Ac4ft-quantifier $\approx^*$. The rule $\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ belongs to the set Ω of relevant G-action rules if it is satisfied $\varphi_{St} \in \mathcal{B}_{A,St}$, $\Phi_{Chg} \in \mathcal{C}_A$, $\psi_{St} \in \mathcal{B}_{S,St}$, $\Psi_{Chg} \in \mathcal{C}_{St}$. An example of input of the *Ac4ft-Miner* procedure is in Fig. 4.



ANTECEDENT STABLE PART

Stable antecedent	1 - 2
» Height(*)	B, pos
» Alcohol(*)	B, pos

Total length: 0 - 99

QUANTIFIERS

Type	Rel.	Value	Units
BASE Before	>=	15.00	Abs.
BASE After	>=	15.00	Abs.
FUI DiffVal	>=	0.40	Abs.

SUCCEDENT STABLE PART

Stable succedent	1 - 1
» Chlst(*)	B, pos

Total length: 0 - 99

(1) ANTECEDENT VARIABLE PART

Variable antecedent	1 - 1
» Bmi(*)	B, pos

CONDITION

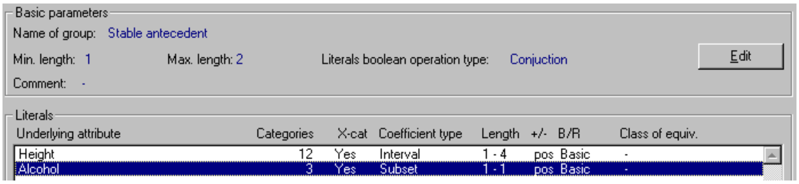
Condition	0 - 99

(2) SUCCEDENT VARIABLE PART

Suc Variable	1 - 1
» Weight(*)	B, pos

Fig. 4. Example of Input of the *Ac4ft-Miner* Procedure

The set $\mathcal{B}_{A,St}$ of relevant Boolean attributes considered as stable antecedents is defined in left upper part named `ANTECEDENT STABLE PART` of Fig. 4, details of the definition are in Fig. 5. The set $\mathcal{B}_{A,St}$ is defined as a conjunction of 1 - 2 of Boolean attributes, see third row in Fig. 5. The conjunction of 1 Boolean attribute is the attribute itself. The Boolean characteristics of the attributes *Height* and *Alcohol* are used. The sets $\mathcal{B}(Height)$ and $\mathcal{B}(Alcohol)$ of relevant Boolean characteristics of the attributes *Height* and *Alcohol* respectively are defined in Fig. 5. The set $\mathcal{B}_{A,St}$ consists of all Boolean attributes φ_1, φ_2, and $\varphi_1 \wedge \varphi_2$ where $\varphi_1 \in \mathcal{B}(Height)$ and $\varphi_2 \in \mathcal{B}(Alcohol)$.



Basic parameters

Name of group: Stable antecedent

Min. length: 1 Max. length: 2 Literals boolean operation type: Conjuction

Comment: -

Literals

Underlying attribute	Categories	X-cat	Coefficient type	Length	+/-	B/R	Class of equiv.
Height	12	Yes	Interval	1 - 4	pos	Basic	-
Alcohol	3	Yes	Subset	1 - 1	pos	Basic	-

Fig. 5. Definition of the set $\mathcal{B}_{A,St}$

The attribute *Height* has 12 categories - intervals of height in cm: $\langle 148; 160)$, $\langle 160; 163)$, ..., $\langle 187; 190)$, $\langle 190; 202)$. The set $\mathcal{B}(Height)$ is defined as the set of all Boolean attributes $Height(\alpha)$ where α is an interval of 1-4 categories (i.e. 1-4 consecutive categories), see expressions `"Interval 1-4"` and `"Coefficient type Length"` in Fig. 5. This way 42 (i.e. 9+10+11+12) particular Boolean attributes are defined. The attribute $Height(\langle 148; 160), \langle 160; 163))$ is an example of Boolean attribute with two consecutive categories - intervals $\langle 148; 160)$

and $\langle 160; 163)$, it is equivalent to *Height*$\langle 148; 163)$. It is true for a patient if the height of this patient is in the interval $\langle 148; 163)$. The attribute *Alcohol* concerns patient's consumption of alcohol. It has 3 categories - *no*, *occasionally*, and *regularly*. The set $\mathcal{B}(Alcohol)$ consists of three Boolean attributes *Alcohol*(*no*), *Alcohol*(*occasionally*), and *Alcohol*(*regularly*). The definition in Fig. 5 means that the set $\mathcal{B}_{A,St}$ consists of 171 Boolean attributes - relevant stable antecedents.

The set $\mathcal{C}_A$ of relevant changes of antecedent is given by a set $\mathcal{B}_{A,Fl}$ of relevant Boolean attributes created from flexible antecedent attributes. Remember the definition of the *change of Boolean attribute* given at the beginning of Sect. 4. The *initial state* $\mathcal{I}(\Lambda)$ of the change of the coefficient $\Lambda = [Z(\kappa) \rightarrow Z(\kappa')]$ is defined as $\mathcal{I}(\Lambda) = Z(\kappa)$ and the *final state* $\mathcal{F}(\Lambda)$ of Λ is defined as $\mathcal{F}(\Lambda) = Z(\kappa')$. The *change of Boolean attribute* is created from *changes of coefficient* and Boolean connectives in the same way as the Boolean attribute is created from the literals. If Φ is a change of Boolean attribute, then the *initial state* $\mathcal{I}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals Λ occurring in Φ by their initial states $\mathcal{I}(\Lambda)$. The *final state* $\mathcal{F}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals Λ occurring in Φ by their final states $\mathcal{F}(\Lambda)$. The set $\mathcal{C}_A$ of relevant changes of antecedent consists of all changes Φ of Boolean attribute such that both $\mathcal{I}(\Lambda) \in \mathcal{B}_{A,Fl}$ and $\mathcal{F}(\Lambda) \in \mathcal{B}_{A,Fl}$.

Note that there are various fine tools how to define a set of relevant Boolean attributes in the procedure *Ac4ft-Miner*. They are the same as in the *4ft-Miner* procedure [10]. Their detailed description is out of the scope of this paper. A simple example is the definition of the set $\mathcal{B}_{A,St}$ above. Other simple examples are the definitions of the sets $\mathcal{B}_{A,Fl}$, $\mathcal{B}_{S,St}$, and $\mathcal{B}_{S,Fl}$ below.

The set $\mathcal{B}_{A,Fl}$ is defined in a left bottom part of Fig. 4 named `ANTECEDENT VARIABLE PART`. Details are not shown here due to limited space. The set $\mathcal{B}_{A,Fl}$ is defined as the set $\mathcal{B}(BMI)$ of Boolean characteristics of the attributes *BMI*, similarly to the definition of the sets $\mathcal{B}(Height)$ above. The attribute *BMI* has 13 categories - intervals $(16; 21\rangle$, $(21; 22\rangle$, ..., $(31; 32\rangle$, > 32. The set $\mathcal{B}(BMI)$ is defined as the set of all Boolean attributes *BMI*(κ) where κ is an interval of 1-3 categories. This way 36 particular Boolean attributes *BMI*(κ) are defined. It means that there are 1010 relevant changes antecedent in the set $\mathcal{C}_A$, all of them have the form $[BMI(\kappa_1) \rightarrow BMI(\kappa_2)]$ where $\kappa_1 \cap \kappa_2 = \emptyset$.

The set $\mathcal{B}_{C,St}$ of relevant Boolean attributes considered as stable succedents is defined in right upper part named `SUCCEDENT STABLE PART` of Fig. 4. The attribute *Chlst* (i.e. Cholesterol in mg%) is used as only one stable succedent attribute and thus the set $\mathcal{B}(Chlst)$ of relevant Boolean characteristics of the attribute *Chlst* corresponds to the set $\mathcal{B}_{C,St}$. The attribute *Chlst* has 19 categories) - intervals ≤ 150, $(150; 160\rangle$, ..., $(310; 320\rangle$, > 320. The set $\mathcal{B}(Chlst)$ is defined as the set of all Boolean attributes *Chlst*(κ) where κ is an interval of 1-4 categories. This way 70 particular Boolean attributes *Chlst*(κ) are defined.

The set $\mathcal{C}_S$ of relevant changes of succedent is defined in right bottom part of Fig. 4 named `SUCCEDENT VARIABLE PART`. The set $\mathcal{B}_{S,Fl}$ of relevant Boolean attributes is used to define $\mathcal{C}_S$ in a same way the set $\mathcal{B}_{A,Fl}$ is used to define $\mathcal{C}_A$, see above. The attribute *Weight* is used as only one flexible succedent attribute and thus the set $\mathcal{B}(\mathit{Weight})$ of relevant Boolean characteristics of the attribute *Weight* corresponds to the set $\mathcal{B}_{S,Fl}$. The attribute *Weight* has 12 categories) - intervals $(50;60\rangle$, $(60;65\rangle$, $\ldots$, $(105;110\rangle$, $(110;135\rangle$. The set $\mathcal{B}(\mathit{Weight})$ is defined as the set of all Boolean attributes $\mathit{Weight}(\kappa)$ where κ is an interval of 1-6 categories. This way 57 particular Boolean attributes $\mathit{Weight}(\kappa)$ are defined.

The Ac4ft-quantifier is defined in the middle upper part of Fig. 4 named `QUANTIFIERS`. There is written `BASE Before >= 15.00`, `BASE After >= 15.00`, and `FUIDiffVal >= 0.40`, thus the Ac4ft-quantifier $\Longrightarrow^{I>F}_{0.4,15,15}$ is used.

There is not known a precise formula to compute the number of relevant changes of antecedents and succedents defined this way, thus it is hard to estimate the number of all relevant G-action rules. The task defined in Fig. 4 was solved in 14 hours and 15 min at PC with 1.33 GHz and 1.99 GB RAM. More than $388 * 10^6$ of action rules $\varphi_{St} \wedge \Phi_{Chg} \Longrightarrow^{I>F}_{0.4,15,15} \psi_{St} \wedge \Psi_{Chg}$ were generated and verified and 30 true rules were found. The strongest one is the rule $\mathcal{R}_0$:

$$\mathit{Height}\langle 175,184) \wedge \mathit{Alcohol}(\mathit{regularly}) \wedge [\mathit{BMI}(21,24\rangle \rightarrow \mathit{BMI}(24,27\rangle] \Longrightarrow^{I>F}_{0.4,15,15}$$

$$\Longrightarrow^{I>F}_{0.4,15,15} \mathit{Cholesterol}(190,230\rangle \wedge [\mathit{Weight}(\leq 80) \rightarrow \mathit{Weight}(80,85\rangle]\ .$$

Its effect can be expressed by two association rules $\mathcal{R}_{0I}$ and $\mathcal{R}_{0F}$ with 4ft-quantifiers $\Rightarrow_{=,0.634,26}$ and $\Rightarrow_{=,0.202,12}$ and 4ft-tables $4ft(\varphi_{0I}, \psi_{0I}, \mathrm{STULONG})$ and $4ft(\varphi_{0F}, \psi_{0F}, \mathrm{STULONG})$ given in Fig. 6.

$$\mathcal{R}_{0I}: \quad \varphi_{0I} \Rightarrow_{=,0.634,26} \psi_{0I} \qquad\qquad \mathcal{R}_{0F}: \quad \varphi_{0F} \Rightarrow_{=,0.202,12} \psi_{0F}$$

Here $\varphi_{0I} = \mathit{Height}\langle 175,184) \ \wedge\ \mathit{Alcohol}(\mathit{regularly}) \ \wedge\ \mathit{BMI}(21,24\rangle$,

STULONG	ψ_{0I}	$\neg\psi_{0I}$
φ_{0I}	26	15
$\neg\varphi_{0I}$	242	1134

$4ft(\varphi_{0I}, \psi_{0I}, \mathrm{STULONG})$

STULONG	ψ_{0I}	$\neg\psi_{0I}$
φ_{0I}	17	67
$\neg\varphi_{0I}$	69	1264

$4ft(\varphi_{0F}, \psi_{0F}, \mathrm{STULONG})$

Fig. 6. $4ft(\varphi_{0I}, \psi_{0I}, \mathrm{STULONG})$ and $4ft(\varphi_{0F}, \psi_{0F}, \mathrm{STULONG})$

$\psi_{0I} = \mathit{Cholesterol}(190,230\rangle \ \wedge\ \mathit{Weight}(\leq 80)$,
$\varphi_{0F} = \mathit{Height}\langle 175,184) \ \wedge\ \mathit{Alcohol}(\mathit{regularly}) \ \wedge\ \mathit{BMI}(24,27\rangle$,
and $\psi_{0F} = \mathit{Cholesterol}(190,230\rangle \wedge \mathit{Weight}(80,85\rangle$.

Note that the strongest rule is the rule with the highest difference of confidences of the rules $\mathcal{R}_{0I}$ and $\mathcal{R}_{0F}$. The found 30 true rules can be grouped into four groups concerning patients with the same height. All results concern patients satisfying *Alcohol*(*regularly*). More detailed interpretation of results requires deeper medical knowledge.

The above described application of the *Ac4ft-Miner* procedure can be understood as an attempt to answer the analytical question *"How to decrease probability of having cholesterol level in a certain interval for patients with height in* *a certain interval and with some type of alcohol consumption by changing BMI and with some induced change of weight?* We can get some variants of answer by analyzing the output of the *Ac4ft-Miner*.

Note that the implementation of the *Ac4ft-Miner* procedure is based on the bit string representation of analyzed data, a-priori algorithm is not used, see Sect. 3. The used algorithm is similar to that of the *4ft-Miner* procedure, see [10]. Its performance is sensitive to the parameters B_1 and B_2 of the Ac4ft-quantifier $\Longrightarrow^{I>F}_{q,B_1,B_2}$. The higher are the B_1 and B_2 the more sure uninteresting G-action rules can be skipped. Let us give overview of results of four runs of the *Ac4ft-Miner* with the sets $\mathcal{B}_{A,St}$, $\mathcal{C}_A$, $\mathcal{B}_{S,St}$, and $\mathcal{C}_S$ specified as above and with different Ac4ft-quantifiers: (1) $\Longrightarrow^{I>F}_{0.4,15,15}$: 14 hours and 15 min, more than $388 * 10^6$ rules really verified and 30 true rules found (see above), (2) $\Longrightarrow^{I>F}_{0.2,50,50}$: 2 hours and 9 min, $33.4 * 10^6$ rules really verified and 216 true rules found, (3) $\Longrightarrow^{I>F}_{0.1,100,100}$: 16 min, 44 sec, $1,81 * 10^6$ rules really verified and no true rule found, (4) $\Longrightarrow^{I>F}_{0.05,200,200}$: 39 sec, 3888 rules really verified and no true rule found.

6 Conclusions

We can conclude that first experiments with the *Ac4ft-Miner* procedure and medical data show results that can be interesting from a medical point of view. The performance of the procedure is reasonable and it can be influenced by dealing with input parameters. The experiments show that the *Ac4ft-Miner* procedure deserves additional research. We suppose namely the following related research activities.

- Experiments with additional Ac4ft-quantifiers. There are various additional Ac4ft-quantifiers inspired by the 4ft-quantifiers used in the applications of the *4ft-Miner* procedure [12]. An example of an interesting additional Ac4ft-quantifier is the Ac4ft-quantifier $\Longrightarrow^{+,F>I}_{q,B_1,B_2}$ defined by the condition

$$\left(\frac{a_I}{a_I+b_I}-\frac{a_I+c_I}{a_I+b_I+c_I+d_I}\right)-\left(\frac{a_F}{a_F+b_F}-\frac{a_I+c_I}{a_I+b_I+c_I+d_I}\right) \geq q \wedge a_I \geq B_1 \wedge a_F \geq B_2$$

  see 4ft-tables in Fig. 2. We assume $0 < q \leq 1$, $B_1 > 0$, and $B_2 > 0$. This Ac4ft-quantifier is inspired by the 4ft-quantifier $\sim^+_{p,Base}$ of *above average dependence* that is for $0 < p$ and $Base > 0$ defined by the condition $\frac{a}{a+b} \geq (1+p)\frac{a+c}{a+b+c+d} \wedge a \geq Base$, see Tab. 1. The rule $\varphi \sim^+_{p,Base} \psi$ means that among objects satisfying φ is at least $100p$ percent more objects satisfying ψ than among all objects and that there are at least $Base$ objects satisfying both φ and ψ [9].
- Detailed study of relation of G-action rules and of the *Ac4ft-Miner* procedure to the approaches described in [5,6,7] that are only shortly mentioned in Sect. 2.
- Study of logic of G-action rules, namely study of correct deduction rules of the form $\frac{\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}}{\varphi'_{St} \wedge \Phi'_{Chg} \approx^* \psi'_{St} \wedge \Psi'_{Chg}}$. We suppose to get results similar to results

on deduction rules of the form $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ [9] concerning association rules $\varphi \approx \psi$ introduced in Sect. 3. Such deduction rules can be used e.g. to optimize the *Ac4ft-Miner* procedure in a similar way the rules $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ [9] are used.
- Possibilities of application of formalized background knowledge stored in the LISp-Miner system. There are possibilities to use such knowledge e.g. to formulate reasonable analytical question and to arrange the output analytical reports [9].
- Research into parallelization of GUHA procedures and using PC-Grid to solve very large tasks.

References

1. Aggraval, R., et al.: Fast Discovery of Association Rules. In: Fayyad, U.M., et al. (eds.) Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park (1996)
2. Hájek, P. (guest editor): International Journal of Man-Machine Studies. special issue on GUHA 10 (1978)
3. Hájek, P., Havránek, T.: Mechanizing Hypothesis Formation (Mathematical Foundations for a General Theory). Springer, Heidelberg (1978)
4. Ras, Z., Wieczorkowska, A.: Action-Rules: How to Increase Profit of a Company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 587–592. Springer, Heidelberg (2000)
5. Ras, Z., Tsay, L.: Discovering the Concise Set of Actionable Patterns. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 169–178. Springer, Heidelberg (2008)
6. Seunghyun, I., Ras, Z.: Action Rule Extraction from a Decision Table: ARED. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 160–168. Springer, Heidelberg (2008)
7. E-Action Rules, System DEAR. In: Lin, T.Y., et al. (eds.) Data Mining: Foundations and Practice. Studies in Computational Intelligence, vol. 118, pp. 289–298. Springer, Heidelberg
8. Rauch, J.: Some Remarks on Computer Realisations of GUHA Procedures. International Journal of Man-Machine Studies 10, 23–28 (1978)
9. Rauch, J.: Logic of Association Rules. Applied Intelligence 22, 9–28 (2005)
10. Rauch, J.: An Alternative Approach to Mining Association Rules. In: Lin, T.Y., et al. (eds.) Data Mining: Foundations, Methods, and Applications, pp. 219–238. Springer, Heidelberg (2005)
11. Rauch, J., Šimunek, M.: GUHA Method and Granular Computing. In: Hu, X., et al. (eds.) Proceedings of IEEE conference Granular Computing (2005)
12. Rauch, J., Tomečková, M.: System of Analytical Questions and Reports on Mining in Health Data – a Case Study. In: Roth, J., et al. (eds.) Proceedings of IADIS European Conference Data Mining 2007, pp. 176–181. IADIS Press (2007)
13. Šimůnek, M.: Academic KDD Project LISp-Miner. In: Abraham, A., et al. (eds.) Advances in Soft Computing - Intelligent Systems Design and Applications. Springer, Heidelberg (2003)