GUHA - a data mining method

David Coufal
Institute of Computer Science AS CR
coufal@cs.cas.cz

GUHA method

(General Unary Hypotheses Automaton)

- a data mining method for automatic search for statistical hypotheses supported by data
- exploratory analysis vs. confirmatory one
- it is based on logical and statistical principles

source database

	clump-th	uc-size	bare-nu	class
object 1	5	1	1	2
object 2	5	4	10	2
object 3	8	10	10	4
object 4	6	8	4	2

- rows – objects
- columns – variables
- cells – values of variables for particular objects

- typical SDs – data tables in Access, Excel, …

Wisconsin Breast Cancer Database

- obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg
- available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- set of 699 cases
- for each case there is observed 10 variables
  9 attributes + 1 classification
- each attribute takes values from {1,2, …10}

part of wbc database

wbcdata.xls

	A	B	C	D	E	F	G	H	I	J
1	clump-th	uc-size	uc-shape	marg-adh	sec-size	bare-nu	bland-ch	norm-nu	mitos	class
2	5	1	1	1	2	1	3	1	1	2
3	5	4	4	5	7	10	3	2	1	2
4	3	1	1	1	2	2	3	1	1	2
5	6	8	8	1	3	4	3	7	1	2
6	4	1	1	3	2	1	3	1	1	2
7	8	10	10	8	7	10	9	7	1	4
8	1	1	1	1	2	10	3	1	1	2
9	2	1	2	1	2	1	3	1	1	2
10	2	1	1	1	2	1	1	1	5	2

wbc

1) clump thickness
2) uniformity of cell size
3) uniformity of cell shape
4) marginal adhesion
5) single epithelial cell size
6) bare nuclei
7) bland chromatin
8) normal nucleoli
9) mitoses
10) class

GUHA procession of source database

source database

⇓

categorization

⇓

(data matrix)

⇓

task definition

⇓

GUHA procession of data

⇓

valid hypotheses

categorization

- categorization is a process of categories creation
- each variable is associated with a set of its categories
- a category is a subset of range of variable

categorization examples

clump-th = {1,2,3,4,5,6,7,8,9,10}
clump-th – clump-th[low], clump-th[mid], clump-th[high]

clump-th[low]={1,2,3},
clump-th[middle]={4,5,6},
clump-th[high]={7,8,9,10}

class = {2,4}
class – class[benign], class[malign]

class[benign]={2}
class[malign]={4}

cedents

- cedent is any combination of categories of form

  C1 & C2 & … & Cn

  where integer n is called as length of cedent

- cedents have a logical interpretation in GUHA

  they are seen as boolean propositions which are evaluated as true or false, verity is denoted by 1 and falsity by 0

- evaluation of simple cedents (categories) → data matrix

src. database	clump-th	uc-size	bare-nu	class
object 1	5	1	1	2
object 2	5	4	10	2
object 3	8	10	10	4
object 4	6	8	4	2

data matrix	clump thickness			uniformity of cell size		
part I	cl-th[low]	cl-th[mid]	cl-th[high]	uc-sz[low]	uc-sz[mid]	uc-sz[high]
object 1	0	1	0	1	0	0
obejct 2	0	1	0	0	1	0
object 3	0	0	1	0	0	1
object 4	0	1	0	0	0	1

data matrix	bare nuclei			class	
part II	br-nu[low]	br-nu[mid]	br-nu[high]	class[benign]	class[malign]
object 1	1	0	0	1	0
obejct 2	0	0	1	1	0
object 3	0	0	1	0	1
object 4	0	1	0	1	0

- evaluation of compound cedents is made according to laws of boolean propositional logic

data matrix	clump thickness			uniformity of cell size		
part I	cl-th[low]	cl-th[mid]	cl-th[high]	uc-sz[low]	uc-sz[mid]	uc-sz[high]
object 1	0	1	1	1	0	0
obejct 2	0	1	0	0	1	0

eval(object2, cl-th[low] & uc-sz[mid] & br-nu[high]) = **0**

data matrix	bare nuclei			class	
part II	br-nu[low]	br-nu[mid]	br-nu[high]	class[benign]	class[malign]
object 1	1	0	0	1	0
obejct 2	0	0	1	1	0

= 0 & 1 & 1 = **0**

evaluation of cedents for all objects

- evaluation of given cedent can be made for each object of source database

- this evaluation gives two numbers

  1) number of object having eval(object, cedent)=1
  i.e., number of objects satisfying given cedent

  2) number of object having eval(object, cedent)=0
  i.e., number of objects not satisfying given cedent

four-fold table

- it is defined for each ordered pair of cedents
  [antecedent - A, succedent - S]

- it is given by evaluation of ordered pair of cedents for given data matrix

	S	**non(S)**
A	a	b
non(A)	c	d

a – number of objects satisfying A and S simultaneously
b – number of objects satisfying A and non(S), simultaneously
c – number of objects not satisfying A and satisfying S, smt.
d – number of objects both not satisfying A and S smt.

hypotheses in GUHA

- general form $A \approx S$
- A,S – antecedent, succedent – some cedents

  each hypothesis determines its four-fold table for given data matrix

- ≈ quantifier – defined by its associated function which is a function evaluating each ff-table by 1 or 0

  if for given hypothesis' ff-table is a value of resp. associated function 1 then we say that this hypothesis is valid in GUHA sense, otherwise it is said to be invalid

FIMPL quantifier - definition

- FIMPL - founded almost implication quantifier $\Rightarrow_{CP,BASE}$

  associated function depends on two parameters cp, base

$$cp \in (0, 1], \quad base \geq 1$$

its value is 1 iff the following holds

$$prob(a, b) = \frac{a}{a + b} \geq cp, \quad a \geq base$$

otherwise it is zero

FIMPL quantifier – explanation

clump-th[high] & bare-nu[>1] $\Rightarrow_{0.9,70}$ class[malign]

	S	non(S)
A	100	5
non(A)	141	453

- cp=0.9, base=70; prob=100/(100+5)=0.95 ≥ 0.9, a=100 ≥ 70
- hypothesis says - conditional probability P(S/A) is greater than cp and value of cases of simultaneous satisfaction of A and S is greater than base
- in other words „many objects satisfying A also satisfy S“

task definition – relevant hypothesis

- relevant hypothesis is given by:

  quantifier selection together with its parameters values
  set of relevant antecedents specification
  set of relevant succedents specification

- relevant (A/S)dent is given by its maximal length and by set of its acceptable categories

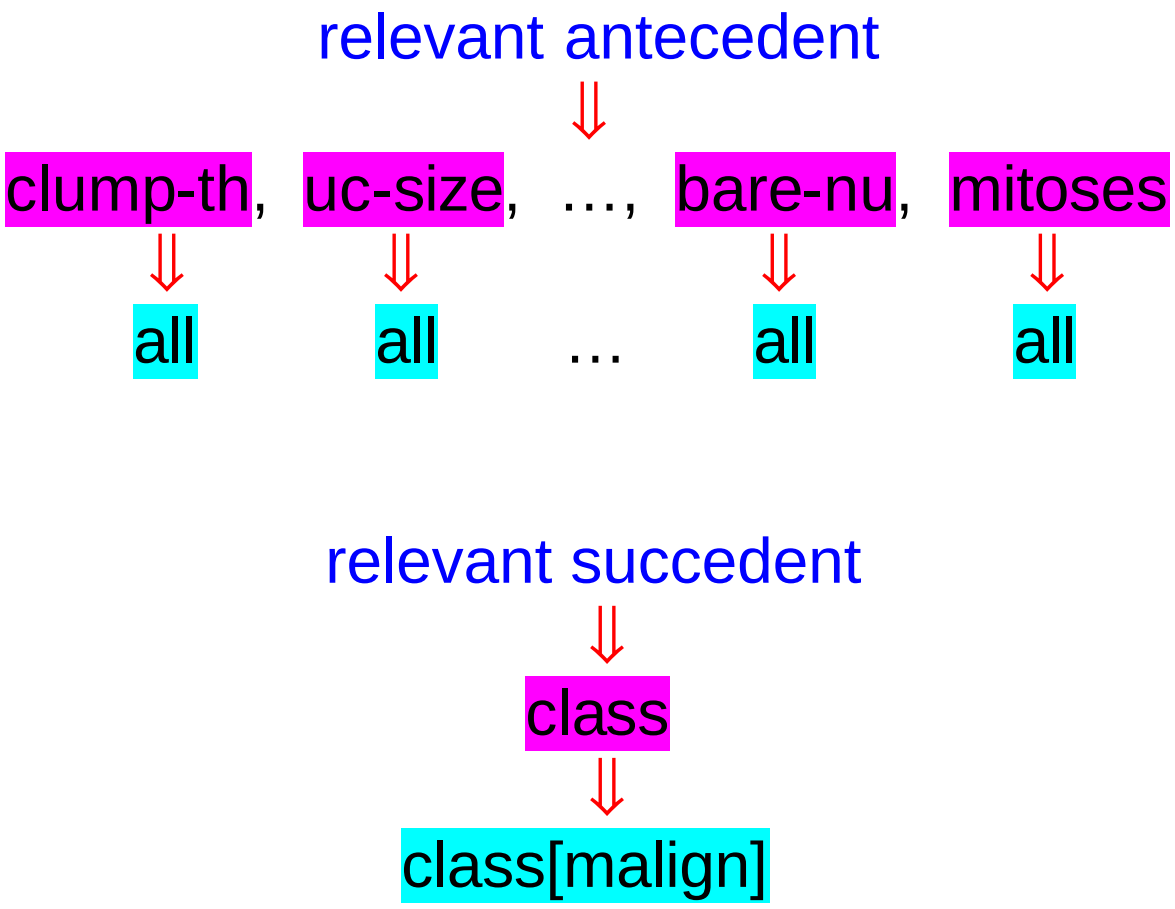
acceptable categories specification

- set of acceptable categories is specified hierarchically by

  1) set of variables whose categories can form relevant cedent

  2) for each of above variables acceptable categories are determined

relevant cedent specification - example



relevant hypotheses testing

- each cedent of length less or equal to maximal length specified and formed from acceptable categories is a relevant (A/S)cedent

- each hypothesis formed from some relevant antecedent and some relevant succedent, together bound by chosen quantifier, is a relevant hypothesis

- in GUHA each relevant hypothesis is tested for its validity, valid hypotheses are revealed

- number of relevant hypotheses is typically extreme combinatorial problem - computerization of the method

GUHA software implementation

- GUHA +– project – implementation of GUHA method on Windows 95/98/NT platform

- three parts:

  1. GUHA Environment – categorization, task definition

  2. GUHA Core – relevant hypotheses generation and testing

  3. GUHA Interpretation – listing, filtering and sorting of valid hypotheses

GUHA +– download

- freeware distribution at

http://www.cs.cas.cz,
section research, part software

conclusion

- GUHA is a datamining method searching for hypotheses supported by data, it is made in an exploratory way

- it is a universal method - no matter from which area data are coming from

- applications in many areas
  e.g., in chemical industry, medicine, banking …