Decision Rules (LERS), Action Rule Discovery

Akhila Sirikonda - 801198913
Keerrthana Sridhar - 801217880
Gopal
Shriya Sharma-801168387
Kamal Mohammed Adil Shaik - 801151613
Hitesh
Raj Shah - 801205036

ACTION RULES

- Action rules are special type of rules, also called actionable rules which can be constructed from classification rules to suggest ways to reclassify objects to desired states.
- In e-commerce applications, this re-classification may mean that a consumer not interested in a certain product, now may buy it, and therefore may fall into a group of more profitable customers. In medical domain, this re-classification may mean how to change the class of a tumor from malignant to benign.
- The groups are formed based on values of classification attributes in decision table schema.By a decision table we mean any information system where the set of attributes is partitioned into conditions and decisions.

ACTION RULES(Continued)

- To discover action rules it is required that the set of conditions is partitioned into stable conditions and flexible conditions/attributes. For simplicity reason, we also assume that there is only one decision attribute.
- For example, *date of birth* is a stable attribute, and *interest rate* on any customer account is a flexible attribute (dependable on bank).
- Actionable rule mining deals with benefit-driven actions required for decision making.
- Actionability is the key concept in most applications because actionable rules allow the user to do his/her job better by taking some specific actions in response to the discovered knowledge.

Decision Table

Any information system S is of the form --

$S = (A_{Fl} \cup A_{St} \cup \{d\})$, where

- d is a distinguished attribute called decision.
- the elements of A_{St} are called stable conditions
- the elements of $A_{Fl} \cup \{d\}$ are called flexible conditions

Example of action rule

$$[(b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \ldots \wedge (b_p, v_p \rightarrow w_p)](x) \Rightarrow [(d, k_1 \rightarrow k_2)](x)$$

Decision Table (Continued)

- This means that, if we change the value of attribute b_1 from v_1 to -> w_1 and the value of attribute b_2 from v_2 to -> w_2 ,and so on, and the value of attribute b_p from v_p to -> w_p , then the value of the decision attribute d, will change from k_1 -> to the desired value k_2.
- Usually, there is a cost (monetary or moral) association with undertaking some kind of an action called cost of Action rule. For example, decreasing the interest rate on a customer account or relocating an employee from one city to another.
- We will denote the cost with p - a number from 0 to $+\infty$. The cost will be close to 0 if the action is trivial (very easy to accomplish) and the cost will be close to plus infinity $+\infty$ if the action is very difficulty (almost impossible) to accomplish

Action Rules

X	a	b	c	d
X1	0	S	0	L
X2	0	R	1	L
X3	0	S	1	L
X4	0	R	1	L
X5	2	P	2	L
X6	2	P	2	L
X7	2	S	2	H

{a, c} - stable attributes,
{b,d} - flexible attributes,
d - decision attribute

(its values are-
L- low profitability customer
H-high profitability customer)

(r_1, r_2)- action rule:
[(b, P® S)](x) Þ[(d, L® H)](x)

Rules Discovered-

r_1 = [(b, P) ® (d, L)]
r_2 = [(a, 2) ^ (b, S) ® (d, H)]

Binding to Thrombin Database

- Thrombin is a serine protease, an enzyme that in humans is encoded by the F2 gene. Drugs are typically small organic molecules that achieve their desired activity by binding to a target site on a receptor. The first step in the discovery of a new drug is usually to identify and isolate the receptor to which it should bind, followed by testing many small molecules for their ability to bind to the target site.

- This leaves researchers with the task of determining what separates the active (binding) compounds from the inactive (non-binding) ones. Such a determination can then be used in the design of new compounds that not only bind, but also have all the other properties required for a drug (solubility, oral absorption, lack of side effects, appropriate duration of action, toxicity, etc.).

Binding to Thrombin database

- The data set consists of 1909 compounds tested for their ability to bind to a target site on thrombin.
- Each compound is described by binary features which describe three-dimensional properties of the molecule. Biological activity in general, and receptor binding affinity in particular, correlate with various structural and physical properties of small organic molecules. The task with KDD Cup 2001 was to determine which of these properties are critical in this case and to learn to accurately predict the class value: Active or Inactive.
- In this testing we use the class attribute, which has value A for active and I for inactive, as the re-classification attribute for the actionRules. In this way, we provide suggestions to the user to what molecular properties can be changed in order to reclassify the chemical compound from inactive to active class, in order to bind to thrombin.

Insurance Company Benchmark Database

- The next database used is in the financial domain, the Insurance Company Benchmark (COIL 2000) database used with the CoIL 2000 Challenge.
- The data contains 5,822 tuples (the customers /rows in the database). The features (the attributes / columns in the database) include product usage data and socio-demographic data derived from zip area codes.
- The data was supplied by the Dutch data mining company Sentient Machine Research and is based on a real world business problem.
- In our testing the user would like to reclassify the attribute Contribution car policies from a value of 5 to 6.

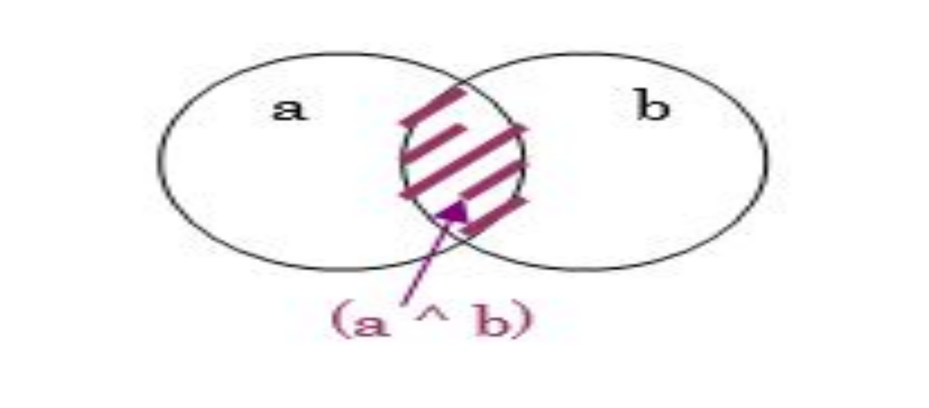
What is LERS ?

- **Stands for Learning from Examples based on Rough Sets**
- It allows us to generate rules, which have a specific attribute on the right hand side of the rule, called as decision attribute.
- Using other attributes (Headache or Temperature) we will come to know whether Flu is there or not.
- LERS Works based on the concept called as **Covering.**

U	Headache	Temp.	Flu
U1	Yes	Normal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	No	High	No
U6	No	Very-high	Yes
U7	No	High	Yes
U8	No	Very-high	No

Covering

- P is a covering of R in S if $P^* \le R^*$ ⌐ ($\exists$Q $\subseteq$ P) [$P^* \le R^*$]
- In other words, P is a covering of R if P is a subset of R , and we do not have a smaller set, which is a subset of R.which implies P is smallest.
- For example, {a} is bigger than {a, b} (and {a, b} is smaller than {a}) because {a} is more general than {a, b}, in other words, {a} includes {a, b}



Steps to Generate LERS Rule

X = {x1,x2,x3,x4,x5,x6,x7,x8} , Classification attribute: {A,B}

Decision Attribute :{C}

Step1: Find Coverings of {A,B,C} and check if coverings of the classification attribute are the subset of covering of decision attribute are the subset of coverings of decision attribute and mark them.

{C,0}* = {x1,x3}; {C,1}* = {x2,x4};

{C,2}* = {x5,x6}; {C,3}* = {x7,x8};

{A,0}* = {x1,x2,x3,x4}; {B,0}* = {x1,x3} $\subseteq$ {C,0}

{A,1}* = {x5,x6} $\subseteq$ {C,2} {B,1}* = {x2,x4,x5,x6};

{A,2}* = {x7,x8} $\subseteq$ {C,3} {C,2}* = {x7,x8} $\subseteq$ {C,3}

X	A	B	C
X1	0	0	0
X2	0	1	1
X3	0	0	0
X4	0	1	1
X5	1	1	2
X6	1	1	2
X7	2	2	3
X8	2	2	3

Steps to Generate LERS Rule

Step 2. Generate rules from the sets in the First loop. *certain rules* are generated from marked sets and *possible rules* are generated from unmarked sets.

Certain rules have a confidence of 100% ,but possible rules have different confidence which is less than 100%.

Certain Rules :

(a,1) → (c,2)

(a,2) → (c,3)

(b,0) → (c,0)

(b,2) → (c,3)

(a,0) &(b,1) → (c,1)

Possible Rules :

(a,0) → (c,0) with confidence = 1/2

(a,0) → (c,1) with confidence = 1/2

(b,1) → (c,1) with confidence = 1/2

(b,1) → (c,2) with confidence = 1/2

Step 3: Next, if there are unmarked sets , then we go to a Second Loop, by combining the unmarked sets together. Continue the process until there are no unmarked sets.

Action Rule Discovery Example

Action rules describe possible transitions of objects from one state to another with respect to a distinguished attribute

Steps to Compose Action Rules

X	A	F	G	C
x1	a2	f1	g3	c2
x2	a1	f2	g1	c1
x3	a1	f2	g2	c1
x4	a1	f1	g1	c2
x5	a2	f2	g2	c2
x6	a1	f2	g3	c2

- Assume C is a decision attribute
- Also, assume that G is a stable attribute
- and {A, F} are flexible.
- Find all action rules re-classifying objects in Table 1 from the class C(c2) to C(c1).

Certain Rules(First Loop)
a2 -> c2
f1 -> c2
g3 -> c2

Certain Rules:(Second Loop)
a1 ^ g2 -> c1
f2 ^ g1 -> c1

-

From:
a2 -> c2
a1 ^ g2 -> c1
we compose the following action rule:
(a, 2->1) ^ (g = 2) -> (c, 2->1)

lly, from:

f1 → c2

f2 ^ g1 → c1

we compose the following action rule:

(f, 1→2) ^ (g = 1) → (c, 2→1)

Thank you!!