Data Analysis - the Rough Sets Perspective

Zdzisław Pawlak

Institute of Computer Science
Warsaw University of Technology
00-665 Warsaw, Nowowiejska 15/19

Abstract: Rough set theory is a new mathematical approach to vagueness and uncertainty The theory has found many interesting applications and can be viewed as a new methodology of data analysis. In this paper basic concepts of rough set theory will be defined and the methodology of applications briefly discussed. A tutorial illustrative example will be used to make the introduced notions more intuitive.

1. Introduction

Rough set theory is a new mathematical approach to vagueness and uncertainty. Practically the theory can be viewed as a new method of data analysis.

Rough set based data analysis starts from a data set organized in a form of a data table. Rows of the table describe objects of interest by means of attribute values. Objects characterized by the same attribute values are *indiscernible*. The indiscernibility relation is the mathematical basis of rough set theory. Any set of all *indiscernible* (*similar*) objects is called an *elementary set*, or *granule* (*atom*) and can be understand a basic knowledge about the universe. Any union of some elementary sets is referred to as a *crisp* (*precise*) set - otherwise the set is *rough* (*imprecise*, *vague*). Each rough set has *boundary-line* cases, i.e., objects which cannot be with certainty classified, by employing the available knowledge, as members of the set or its complement. Thus rough sets, in contrast to precise sets, cannot be characterized in terms of information about their elements.

With any rough set a pair of precise sets - called the *lower* and the *upper approximation* of the rough set is associated. The lower approximation consists of all objects which *surely* belong to the set and the upper approximation contains all objects which *possibly* belong to the set. The difference between the upper and the lower approximation constitutes the *boundary region* of the rough set. Approximations are basic operations in rough set theory.

Rough set theory seems to be of fundamental importance to AI and cognitive sciences, especially in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, inductive reasoning and pattern recognition. It seems of particular importance to data analysis.

Rough set theory has been applied to many real-life problems in medicine, pharmacology, engineering, banking, financial and market analysis, and others.

Rough set approach to data analysis has many important advantages. Some of them are listed below.

– Provides efficient algorithms for finding hidden patterns in data.

- Finds minimal sets of data (data reduction).
- Evaluates significance of data.
- Generates sets of decision rules from data.
- It is easy to understand.
- Offers straightforward interpretation of obtained results.

This paper gives rudiments of rough set theory. The basic concepts of the theory are illustrated by a simple tutorial example. In order to tackle many sophisticated real life problems the theory has been generalized in various ways but we will not discuss these generalizations here.

Rudiments of the theory can be found in [1, 2, 3]. Various applications and extension of the theory are discussed in [5, 6, 7, 8].

2. Information systems

Starting point of rough set based data analysis is a data set, called an information system.
An information system is a data table, whose columns are labeled by attributes, rows are labeled by objects of interest and entries of the table are attribute values.

Formally, by an *information system* is a pair $S = (U, A)$, where U and A, are finite, nonempty sets called the *universe*, and the set of *attributes*, respectively. With every attribute $a \in A$ we associate a set V_a, of its *values*, called the *domain* of a. Any subset B of A determines a binary relation $I(B)$ on U, which will be called an *indiscernibility relation*, and is defined as follows: $(x, y) \in I(B)$ if and only if $a(x) = a(y)$ for every $a \in A$, where $a(x)$ denotes the value of attribute a for element x. Obviously $I(B)$ is an equivalence relation. The family of all equivalence classes of $I(B)$, i.e., a partition determined by B, will be denoted by $U/I(B)$, or simply by U/B; an equivalence class of $I(B)$, i.e., block of the partition U/B, containing x will be denoted by $B(x)$.

If (x, y) belongs to $I(B)$ we will say that x and y are *B-indiscernible* (*indiscernible with respect to B*). Equivalence classes of the relation $I(B)$ (or blocks of the partition U/B) are referred to as *B-elementary sets* or *B-granules*.

If we distinguish in an information system two disjoint classes of attributes, called *condition* and *decision attributes*, respectively, then the system will be called a *decision table* and will be denoted by $S = (U, C, D)$, where C and D are disjoint sets of condition and decision attributes, respectively.

An example of an information system is shown in Table 1.

Table 1: An example of an information system

F	driving conditions			consequence	support
	weather	*road*	*time*	*accident*	
1	*misty*	*icy*	*day*	*yes*	6
2	*foggy*	*icy*	*night*	*yes*	8
3	*misty*	*not icy*	*night*	*yes*	5
4	*sunny*	*icy*	*day*	*no*	55
5	*foggy*	*not icy*	*dusk*	*yes*	11
6	*misty*	*not icy*	*night*	*no*	15

In Table 1 six facts concerning a hundred cases of driving a car in various weather conditions are presented. In the table *weather*, *road* and *time*, are *condition attributes*, represent driving conditions, *accident*, is a decision attribute, giving information whether an accident has occurred or not. *N* is the number of similar cases.

Each row of the decision table determines a decision obeyed when specified conditions are satisfied.

3. Approximation

Suppose we are given an information system $S = (U, A)$, $X \subseteq U$, and $B \subseteq A$. Our task is to describe the set X in terms of attribute values from B. To this end we define two operations assigning to every $X \subseteq U$ two sets $B_*(X)$ and $B^*(X)$ called the *B-lower* and the *B-upper approximation* of X, respectively, and defined as follows:

$$B_*(X) = \bigcup_{x \in U} \{B(x) : B(x) \subseteq X\}, \tag{1}$$

$$B^*(X) = \bigcup_{x \in U} \{B(x) : B(x) \cap X \neq \emptyset\}. \tag{2}$$

Hence, the B-lower approximation of a set is the union of all B-granules that are included in the set, whereas the *B-upper* approximation of a set is the union of all B-granules that have a nonempty intersection with the set. The set

$$BN_B(X) = B^*(X) - B_*(X) \tag{3}$$

will be referred to as the *B-boundary region* of X.

If the boundary region of X is the empty set, i.e., $BN_B(X) = \emptyset$, then X is *crisp* (*exact*) with respect to B; in the opposite case, i.e., if $BN_B(X) \neq \emptyset$, X is referred to as *rough* (*inexact*) with respect to B.

For example, the lower approximation of the set of accident {1, 2, 3, 5} is the set {1, 2, 5}; the upper approximation of this set is the set {1, 2, 3, 5, 6}, and the boundary region is the set {3, 6}.

4. Dependency of attributes and reducts

Important concept in rough set theory is the dependency between attributes. Intuitively, a set of attributes *D* depends totally on a set of attributes *C*, denoted $C \Rightarrow D$, if all values of attributes from *D* are uniquely determined by values of attributes from *C*. In other words, *D* depends totally on *C*, if there exists a functional dependency between values of *C* and *D*.

Formally dependency can be defined in the following way. Let *D* and *C* be subsets of *A*.

We will say that *D depends on C* in the *degree k* ($0 \le k \le 1$), denoted $C \Rightarrow_k D$, if

$$k = \gamma(C,D) = \frac{card(POS_C(D))}{card(U)}, \tag{4}$$

where

$$POS_C(D) = \bigcup_{X \in U/D} C_*(X), \tag{5}$$

called a *positive region* of the partition *U*/*D* with respect to *C*, is the set of all elements of *U* that can be uniquely classified to blocks of the partition *U*/*D*, by means of *C*.

Obviously

$$\gamma(C,D) = \sum_{X \in U/D} \frac{card(C_*(X))}{card(U)}. \tag{6}$$

If $k = 1$ we say that *D depends totally* on *C*, and if $k < 1$, we say that *D depends partially* (in the *degree k*) on *C*. The degree of dependency γ (*C*, *D*) can be also understood as a *consistency measure* of the decision table $S = (U, C, D)$.

The coefficient *k* expresses the ratio of all elements of the universe, which can be properly classified to blocks of the partition *U*/*D*, employing attributes *C* and will be called the *degree of the dependency*.

Important task in rough set based data analysis is data reduction.

Let *C*, $D \subseteq A$, be sets of condition and decision attributes respectively. We will say that $C' \subseteq C$ is a *D-reduct* (reduct with *respect*} to *D*) of *C*, if *C'* is a minimal subset of *C* such that

$$\gamma(C,D) = \gamma(C',D). \tag{7}$$

The intersection of all *D*-reducts is called a *D-core* (core with *respect* to *D*).

Because the core is the intersection of all reducts, it is included in every reduct, i.e., each element of the core belongs to some reduct. Thus, the non-empty core consists of the most important subset of attributes, for none of its elements can be removed without affecting the classification power of attributes.

It is easy to compute that there are two reducts {*weather, road*} and {*weather, time*} of condition attributes in the information system given in Table 1. Thus the core is the attribute *weather*.

5. Decision rules

In this section we define a formal language which will be used for description of properties of data in a data table. The language will consist of implications in the form "*if ... then*..." called decision rules. Decision rules are closely associated will approximations, which will be discussed in the sequel.

Let $S = (U, A)$ be an information system. With every $B \subseteq A$ we associate a formal language, i.e., a set of formulas $For(B)$. Formulas of $For(B)$ are built up from attribute-value pairs (a, v) where $a \in B$ and $v \in V_a$ by means of logical connectives $\wedge$ (*and*), $\vee$ (*or*), $\sim$(*not*) in the standard way.

For any $\Phi \in For(B)$ by $||\Phi||_S$ we denote the set of all objects $x \in U$ satisfying Φ in S and refer to as the *meaning* of Φ in S.

The meaning $||\Phi||_S$ of Φ in S is defined inductively as follows: $||(a, v)||_S = \{x \in U : a(v) = x\}$ for all $a \in B$ and $v \in V_a$, $||\Phi \vee \Psi||_S = ||\Phi||_S \cup ||\Psi||_S$, $||\Phi \wedge \Psi||_S = ||\Phi||_S \cap ||\Psi||_S$, $||\sim \Phi||_S = U - ||\Phi||_S$.

A formula Φ is *true* in S if $||\Phi||_S = U$.

A *decision rule* in S is an expression $\Phi \to \Psi$, read *if* Φ *then* Ψ, where $\Phi \in For(C)$, $\Psi \in For(D)$ and C, D are condition and decision attributes, respectively; Φ and Ψ are referred to as *conditions* and *decisions* of the rule, respectively.

card $(||\Phi \wedge \Psi||_S)$ will be called the *support* of the rule in S and will be denoted by $supp_S(\Phi, \Psi)$.

The number

$$\sigma_S(\Phi, \Psi) = \frac{supp_S(\Phi, \Psi)}{card(U)} \tag{8}$$

will be called the *strength* of the decision rule $\Phi \to \Psi$ in S.

A decision rule $\Phi \to \Psi$ is *true* in S if $||\Phi||_S \subseteq ||\Psi||_S$.

We consider a probability distribution $p_U(x)=1/card(U)$ for $x \in U$ where U is the (non-empty) universe of objects of S; we have $p_U(X) = card(X)/card(U)$ for $X \subseteq U$. For any formula Φ we associate its probability in S defined by

$$\pi_S(\Phi) = p_U(\|\Phi\|_S). \tag{9}$$

With every decision rule $\Phi \to \Psi$ we associate a conditional probability

$$\pi_S(\Psi|\Phi) = p_U(\|\Psi\|_S \mid \|\Phi\|_S) \tag{10}$$

called the *certainty factor* of the decision rule, denoted $cer_S(\Phi, \Psi)$ to estimate the probability of implications. We have

$$cer_S(\Phi, \Psi) = \pi_S(\Psi|\Phi) = \frac{card(\|\Phi \wedge \Psi\|_S)}{card(\|\Phi\|_S)} \tag{11}$$

where $\|\Phi\|_S \neq \emptyset$.

This coefficient is now widely used in data mining and is called *confidence coefficient*.

Obviously, $cer_S(\Phi, \Psi) = 1$ if and only if $\Phi \to \Psi$ is true in S.

If $cer_S(\Phi, \Psi) = 1$, then $\Phi \to \Psi$ will be called a *certain decision* rule; if $0 < cer_S(\Phi, \Psi) < 1$ the decision rule will be referred to as a *uncertain decision* rule.

Besides, we will also use a *coverage factor* of the decision rule, denoted $cov_S(\Phi, \Psi)$, and defined by

$$\pi_S(\Phi|\Psi) = p_U(\|\Phi\|_S \mid \|\Psi\|_S). \tag{12}$$

Obviously we have

$$cov_S(\Phi, \Psi) = \pi_S(\Phi|\Psi) = \frac{card(\|\Phi \wedge \Psi\|_S)}{card(\|\Psi\|_S)}. \tag{13}$$

The certainty factors in S can be interpreted as the frequency of objects having the property Ψ in the set of objects having the property Φ and the coverage factor - as the frequency of objects having the property Φ in the set of objects having the property Ψ.

That the certain decision rules are associated with the lower approximation, whereas the possible decision rules correspond to the boundary region of a set.

For example,

- certain rules describing accidents i.e., the lower approximation of the set of facts $\{1, 2, 3, 5\}$ with certainty 1.00

  1) (*weather*, *misty*) $\wedge$ (*road, icy*) $\to$ (*accident*, *yes*)

  2) (*weather*, *foggy*) $\to$ (*accident*, *yes*)

- uncertain rule describing accidents, i.e., the boundary region $\{3, 6\}$ of the set of facts $\{1, 2, 3, 5\}$ with certainty 0.25

  3) (*weather*, *misty*}) $\wedge$ (*road*, *not icy*}) $\to$ (*accident*, *yes*)

- certain rule describing lack of accidents , i.e., the lower approximation of the set of facts $\{4, 6\}$ with certainty 1.00

  4) (*weather*, *sunny*) $\to$ (*accident*, *no*)

- uncertain rule describing lack of accidents, i.e., the boundary region $\{3, 6\}$ of the set of facts $\{4, 6\}$ with certainty 0.75

  5) (*weather*, *misty*) $\wedge$ (*road*, *not icy*) $\to$ (*accident*, *no*)

6. Decision algorithm

A set of decision rules satisfying some conditions, discussed in what follows, will be called a decision algorithm. Decision algorithm is a logical counterpart of a decision table, and the conditions, that must be satisfied by decision rules in the decision algorithm are chosen in such a way that basic properties of the decision table are preserved.

Let $Dec(S) = \{\Phi_i \rightarrow \Psi_i\}_{i=1}^{m}$, $m \geq 2$, be a set of decision rules in a decision table $S = (U, C, D)$.

1) If for every $\Phi \rightarrow \Psi$, $\Phi' \rightarrow \Psi' \in Dec(S)$ we have $\Phi = \Phi'$ or $\|\Phi \wedge \Phi'\|_S = \emptyset$, and $\Psi = \Psi'$ or $\|\Psi \wedge \Psi'\|_S = \emptyset$, then we will say that *Dec*(*S*) is the set of *mutually disjoint* (*independent*) decision rules in *S*.
2) If $\|\bigvee_{i=1}^{m} \Phi_i\|_S = U$ and $\|\bigvee_{i=1}^{m} \Psi_i\|_S = U$ we will say that the set of decision rules *Dec*(*S*) *covers U*.
3) If $\Phi \rightarrow \Psi \in Dec(S)$ and $\|\Phi \wedge \Psi\|_S \neq \emptyset$ we will say that the decision rule $\Phi \rightarrow \Psi$ is *admissible* in *S*.
4) If $\bigcup_{X \in U/D} C_*(X) = \|\bigvee_{\Phi \in Dec^+(S)} \Phi\|_S$ where $Dec^+(S)$ is the set of all certain decision rules from *Dec* (*S*), we will say that the set of decision rules *Dec* (*S*) preserves the *consistency* of the decision table S = (*U*, *C*, *D*).

The set of decision rules *Dec* (*S*) that satisfies 1), 2), 3) and 4), i.e., is independent, covers *U*, preserves the consistency of *S* and all decision rules $\Phi \rightarrow \Psi \in Dec(S)$ are admissible in *S* - will be called a *decision algorithm* in *S*.

Hence, if *Dec* (*S*) is a decision algorithm in *S* then the conditions of rules from *Dec* (*S*) define in *S* a partition of *U*. Moreover, the *positive region of D with respect to C*, i.e., the set

$$\bigcup_{X \in U/D} C_*(X) \tag{14}$$

is partitioned by the conditions of some of these rules, which are certain in *S*.

An example of decision algorithm associated with Table 1 is given below:

1) (*weather*, *misty*) ∧ (*road*, *icy*) → (*accident*, *yes*)
2) (*weather*, *foggy*) → (*accident*, *yes*)
3) (*weather*, *misty*) ∧ (*road*, *not icy*) → (*accident*, *yes*)
4) (*weather*, *sunny*) → (*accident*, *no*)
5) (*weather*, *misty*) ∧ (*road*, *not icy*) → (*accident*, *no*)

If $\Phi \rightarrow \Psi$ is a decision rule then the decision rule $\Psi \rightarrow \Phi$ will be called an *inverse* decision rule of $\Phi \rightarrow \Psi$.

$Dec^*(S)$ the set of all inverse decision rules of *Dec* (*S*) will be called an inverse decision algorithm of *Dec* (*S*).

For example, the following is the inverse decision algorithm of the decision algorithm 1)-5):

1') (*accident*, *yes*) → (*road*, *icy*}) ∧ (*weather*, *misty*)

2') (*accident*, *yes*) → (*weather*, *foggy*)

3') (*accident*, *yes*) → (*road, not icy*) ∧ (*weather*, *misty*)

4') (*accident*, *no*) → (*weather*, *sunny*)

5') (*accident*, *no*) → (*road*, *not icy*) ∧ (*weather*, *misty*)

7. Properties of decision algorithms

Decision algorithms have interesting probabilistic properties which are discussed in this section after [4].

Let *Dec*(*S*) be a decision algorithm and let Φ → Ψ ∈ *Dec*(*S*). Then the following properties are valid:

$$\sum_{\Phi' \in C(\Psi)} cov_S(\Phi', \Psi) = 1 \tag{15}$$

$$\sum_{\Psi' \in C(\Phi)} cer_S(\Phi, \Psi') = 1 \tag{16}$$

$$\pi_S(\Psi) = \sum_{\Phi' \in C(\Psi)} cer_S(\Phi', \Psi) \cdot \pi_S(\Phi') = \sum_{\Phi' \in C(\Psi)} \sigma_S(\Phi', \Psi) \tag{17}$$

$$cov_S(\Phi, \Psi) = \frac{cer_S(\Phi, \Psi) \cdot \pi_S(\Phi)}{\sum_{\Phi' \in C(\Psi)} cer_S(\Phi', \Psi) \cdot \pi_S(\Phi')} = \frac{\sigma_S(\Phi, \Psi)}{\pi_S(\Psi)} \tag{18}$$

That is, any decision algorithm, and consequently any decision table, satisfies 15)-18). Observe that (17) is the well known *total probability theorem* and (18) is the *Bayes*' *theorem*.

For example, the certainty and coverage factors for decision rules 1)-5) are presented in Table 2.

Table 2.Certainty and coverage factors

rule no.	*certainty*	*coverage*	*support*	*strength*
1	1.00	0.20	6	0.06
2	1.00	0.63	19	0.19
3	0.25	0.17	5	0.05
4	1.00	0.79	55	0.55
5	0.75	0.21	15	0.15

Summing up, the decision algorithm leads to the following conclusions:

– *misty weather and icy road or foggy weather surely cause accident*,

– *misty weather and not icy road most probably does not cause accident*,

– *sunny weather surely does not causes accident.*

The inverse decision algorithm gives the following explanations:

– *if accident occured then the weather was most probably foggy*,

– *if accident does not occured then the weather was surely sunny*.

Note that we are not referring to prior and posterior probabilities - fundamental in Bayesian data analysis philosophy.

In order to compute the certainty and coverage factors of decision rules according to formula (18) it is enough to know the strength (support) of all decision rules in the decision algorithm only. The strength of decision rules can be computed from the data or can be a subjective assessment.

8. Conclusions

In the paper basic concepts of rough set theory have been presented and briefly discussed. The theory attracted attention both theoreticians and practitioners world wide and proved to be a valuable new method of data analysis.

References

[1] Duentsch, I., Guenter, G. Rough set data analysis - a road to non-invasive knowledge discovery. Metoδos Publisher, Bangov, Bissendorf, 2000.

[2] Grzymala-Busse, J. Managing Uncertainty in Expert Systems. Kluwer Academic Publishers, Boston, Dordrecht, 1991.

[3] Pawlak, Z. Rough Sets - Theoretical Aspects of Reasoning about Data; Kluwer Academic Publishers: Boston, Dordrecht, 1991.

[4] Pawlak, Z. Rough Set and Decision Algorithms (to appear).

[5] Polkowski, L., Skowron, A. (eds.) Proceedings of the First International Conference Rough Sets and Current Trends in Computing (RSCTC'98), Warsaw, Poland, June, Lecture Notes in Artificial Intelligence 1424, Springer--Verlag, Berlin, 1998.

[6] Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery Vol. 1-2, Physica-Verlag, Heidelberg, 1998.

[7] Polkowski, L., Tsumoto, S., Lin , T. Y. (eds) Rough Set Method and Applications. New Development in Knowledge Discovery in Information Systems. Physica-Verlag, Heidelberg, (to appear)

[8] Zhong, N., Skowron, A., Ohsuga, S. (eds.) Proceedings of 7th International Workshop: New Directions in Rough Sets, Data Mining, and Granular - Soft Computing (RSFDGSC'99), Yamaguchi, Japan, November 1999, Lecture Notes in Artificial Intelligence 1711 Springer-Verlag, Berlin, 1999, 1-9.

MORE INFO ABOUT ROUGH SETS CAN BE FOUND IN:

http://www.cs.uregina.ca/~roughset

http://www.infj.ulst.ac.uk /staff/I.Duentsch http://www-idss.cs.put.poznan.pl/staff/slowinski/

http://alfa/logic/

http://www.idi.ntnu.no/~aleks/rosetta/

http://www.infj.ulst.ac.uk/~cccz23/grobian/grobian.html

ROUGH SET SOFTWARE

LERS - A Knowledge Discovery System
Jerzy W. Grzymala-Busse
University of Kansas, Lawrence, KS 66045, USA, e-mail: jerzy@eecs.ukans.edu

RSDM: Rough Sets Data Miner, A System to add Data Mining Capabilities to RDBMS
Maria C. Fernandez-Baizan, Ernestina Menasalvas Ruiz, e-mail: cfbaizan@fi.upm.es

ROSETTA
http://www.idi.ntnu.no/~aleks/rosetta.html
Andrzej Skowron
Institute of Mathematics, University of Warsaw , Banacha 2, 02-097 Warsaw, Poland
e-mail:skowron@mimuw.edu.pl

GROBIAN
http://www.infj.ulst.ac.uk/~cccz23/grobian/grobian.html
Ivo Düntsch, Günther Gediga
School of Information and Software Engineering, University of Ulster, Newtownabbey, BT 37 0QB, N.Ireland, e-mail: I.Duentsch@ulst.ac.uk
FB Psychologie Methodenlehre, Universit(t Osnabr(ck, 49069 Osnabr(ck, Germany,
e-mail: ggediga@luce.psycho.Uni-Osnabrueck.DE

TRANCE: a Tool for Rough Data Analysis, Classification, and Clustering
Wojciech Kowalczyk
Vrije Universiteit Amsterdam De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands
e-mail:wojtek@cs.vu.nl

PROBROUGH - A System for Probabilistic Rough Classifiers Generation
Andrzej Lenarcik, Zdzisław Piasta
Kielce University of Technology, Mathematics Department, Al. 1000-lecia P.P. 5, 25-314 Kielce, Poland, e-mail: {zpiasta,lenarcik}@sabat.tu.kielce.pl

ROUGH FAMILY - Software Implementation of the Rough Set Theory
Roman Słowiński and Jerzy Stefanowski
Institute of Computing Science, Poznań University of Technology, 3A Piotrowo Street, 60-965 Poznań, Poland, e-mail: Roman.Slowinski@cs.put.poznan.pl, Jerzy.Stefanowski@cs.put.poznan.pl

TAS: Tools for Analysis and Synthesis of Concurrent Processes using Rough Set Methods
Zbigniew Suraj
Institute of Mathematics, Pedagogical University, Rejtana 16A, 35-310 Rzeszów, Poland
e-mail: zsuraj@univ.rzeszow.pl

ROUGHFUZZYLAB - a System for Data Mining and Rough and Fuzzy Sets Based Classification
Roman W. Swiniarski
San Diego State University, San Diego, California 92182-7720, U.S.A.

PRIMEROSE
Shusaku Tsumoto
Medical Research Institute, Tokyo Medical and Dental University, 1-5-45 Yushima, Bunkyo-ku Tokyo 113 Japan, e-mail: tsumoto@computer.org

KDD-R: Rough Sets-Based Data Mining System
Wojciech Ziarko
Computer Science Department, University of Regina, Regina Saskatchewan, S4S-0A2, Canada, e-mail: ziarko@cs.uregina.ca