Classifiers Based on Approximate Reasoning Schemes

Jan Bazan[1] and Andrzej Skowron[2]

[1] Institute of Mathematics, University of Rzeszów
Rejtana 16A, 35-959 Rzeszów, Poland
bazan@univ.rzeszow.pl
[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

Summary. We discuss classifiers [3] for complex concepts constructed from data sets and domain knowledge using *approximate reasoning schemes* (AR schemes). The approach is based on granular computing methods developed using rough set and rough mereological approaches [9, 13, 7]. In experiments we use a road simulator (see [15]) making it possible to collect data, e.g., on vehicle-agents movement on the road, at the crossroads, and data from different sensor-agents. We compare the quality of two classifiers: the standard rough set classifier based on the set of minimal decision rules and the classifier based on AR schemes.

1 Introduction

A *classification algorithm* (*classifier*) permits making a forecast in new situations on the basis of accumulated knowledge. We consider here classifiers predicting decisions for objects previously unseen; each new object will be assigned to a class belonging to a predefined set of classes on the basis of observed values of suitably chosen attributes (features).

Many approaches have been proposed for constructing of classifiers. Among them we would like to mention classical and modern statistical techniques, neural networks, decision trees, decision rules and inductive logic programming (see e.g. [5] for more details).

One of the most popular methods for classification algorithms constructing is based on learning rules from examples. The standard rough set methods based on calculation of so called *local reducts* makes it possible to compute, for a given data, the descriptions of concepts by means of minimal consistent decision rules (see, e.g., [6], [2]).

Searching for relevant patterns for complex concepts can be performed using AR schemes. AR schemes (see, e.g., [13]) can be treated as approximations of reasoning performed on concepts from domain knowledge and they represent relevant patterns for complex classifier construction. The proposed approach is based on granular computing methods developed using rough set and rough mereological approaches [9, 13, 7].

In our experiments we use a road simulator (see [15]) making it possible to collect data, e.g., on vehicle-agents movement on the road and at the cross-roads and data from different sensor-agents. The simulator also registers a few more features, whose values are defined by an expert.

Any AR scheme is constructed from labelled approximate rules, called *productions* that can be extracted from data using domain knowledge [13]. In the paper we present a method for extracting productions from data collected by road simulator and an algorithm for classifying objects by productions, that can be treated as an algorithm for on-line synthesis of AR scheme for any tested object.

We report experiments supporting our hypothesis that classifiers induced using the AR schemes are of higher quality than the traditional rough set classifiers (see Section 5). For comparison we use data sets generated by road simulator.

2 Approximate reasoning scheme

One of the main tasks of data exploration [4] is discovery from available data and expert knowledge of concept approximations expressing properties of the investigated objects and rules expressing dependencies between concepts. Approximation of a given concept can be constructed using relevant patterns. Any such pattern describes a set of objects belonging to the concept to a degree p where $0 < p < 1$.

Relevant patterns for complex concepts can be represented by AR schemes. AR schemes can be treated as approximations of reasoning performed on concepts from domain knowledge.

Any AR scheme is constructed from labeled approximate rules, called productions. Productions can be extracted from data using domain knowledge. We define productions as a parameterized implications with premises and conclusion built from patterns sufficiently included in the approximated concept.

In Figure 1 we present an example of production for some concepts *C1*, *C2* and *C3* approximated by three linearly ordered layers *small*, *medium*, and *large*. This production is a collection of three simpler rules, called *production rules*, with the following interpretation: (1) if inclusion degree to a concept *C1* is at least *medium* and to concept *C2* at least *large* then the inclusion degree to a concept *C3* is at least *large*; (2) if the inclusion degree to a concept *C1* is at least *small* and to a concept *C2* at least *medium* then the inclusion degree to a concept *C3* is at least *medium*; (3) if the inclusion degree to a concept *C1* is at least *small* and to a concept *C2* at least *small* then the inclusion degree to a concept *C3* is at least *small*.

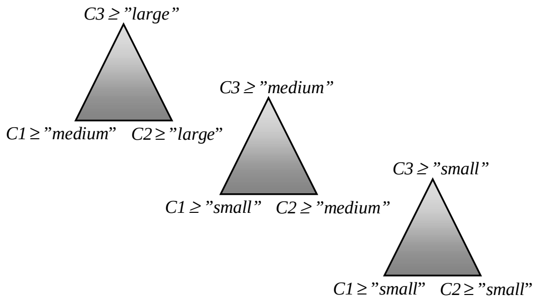


Fig. 1. The example of production as a collection of three production rules

The concept from the upper level of production is called the target concept of production, whilst the concept from the lower level of production are called the source concepts of production. For example, in case of production from Figure 1 *C3* is the target concept and *C1*, *C2* are the source concepts.

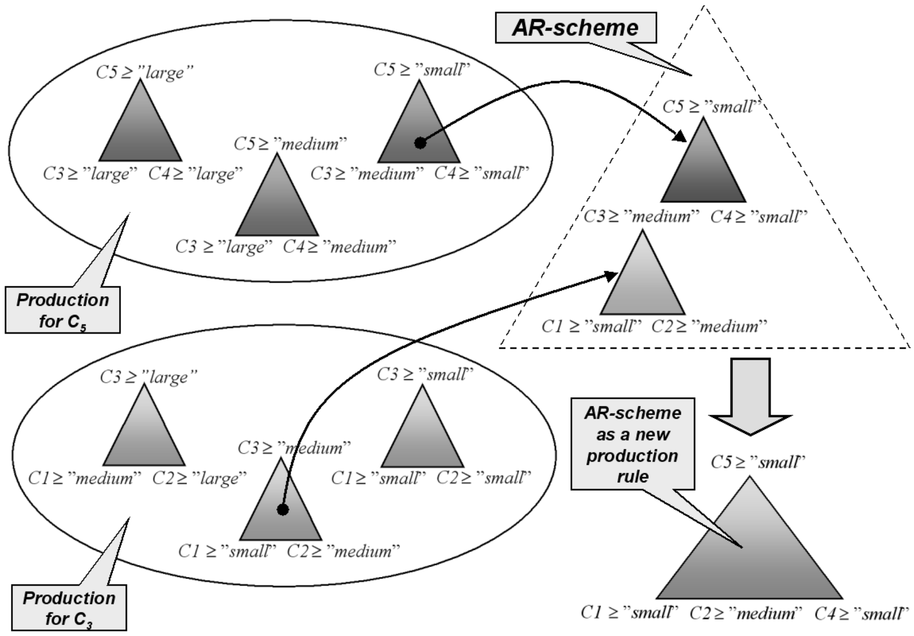


Fig. 2. Synthesis of approximate reasoning scheme

One can construct AR scheme by composing single production rules chosen from different productions from a family of productions for various target concepts. In Figure 2 we have two productions. The target concept of the first production is *C5* and the target concept of the second production is the concept *C3*. We select one production rule from the first production and one production rule from the second production. These production rules are composed and then a simple AR-scheme is obtained that can be treated as a new two-levels production rule. Notice, that the target pattern of lower production rule in this AR-scheme is the same as one of the source patterns from the higher production rule. In this case, the common pattern is described as follows: inclusion degree (of some pattern) to a concept *C3* is at least *medium*.

In this way, we can compose AR-schemes into hierarchical and multilevel structures using productions constructed for various concepts.

3 Road simulator

Road simulator (see [15]) is a tool for generating data sets recording vehicle movement on the road and at the crossroads (see [15]). Such data is extremely crucial in testing of complex decision systems monitoring the situation on the road that are working on the basis of information coming from different devices.

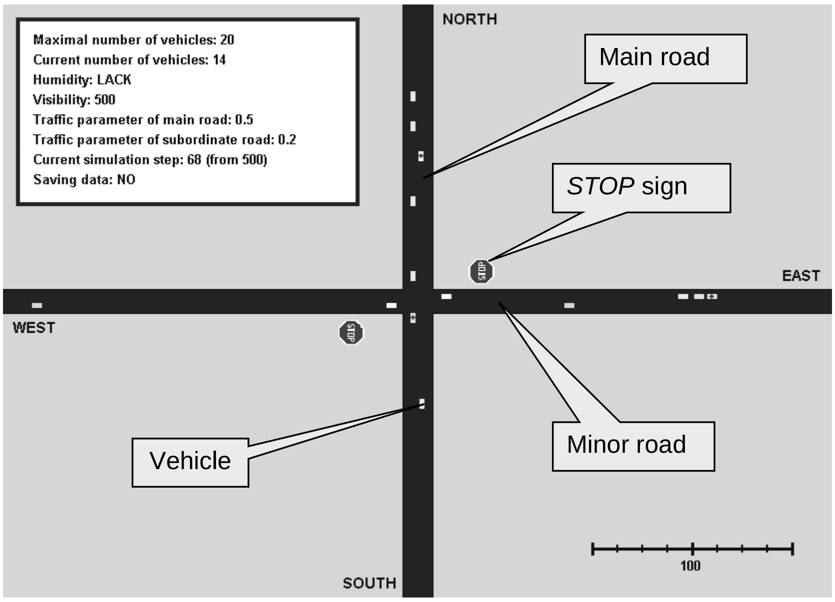


Fig. 3. The board of simulation

Driving simulation takes place on a board (see Figure 3) which presents a crossroads together with access roads.

During the simulation the vehicles may enter the board from all four directions that is East, West, North and South. The vehicles coming to the crossroads form South and North have the right of way in relation to the vehicles coming from West and East.

Each of the vehicles entering the board has only one aim - to drive through the crossroads safely and leave the board. The simulation takes place step by step and during each of its steps the vehicles may perform the following maneuvers during the simulation: passing, overtaking, changing direction (at the crossroads), changing lane, entering the traffic from the minor road into the main road, stopping and pulling out.

Planning each vehicle's further steps takes place independently in each step of the simulation. Each vehicle, is "observing" the surrounding situation on the road, keeping in mind its destination and its own parameters (driver's profile), makes an independent decision about its further steps; whether it should accelerate, decelerate and what (if any) maneuver should be commenced, continued, ended or stopped.

Making decisions concerning further driving, a given vehicle takes under consideration its parameters and the driving parameters of five vehicles next to it which are marked FR1, FR2, FL, BR and BL (see Figure 4).

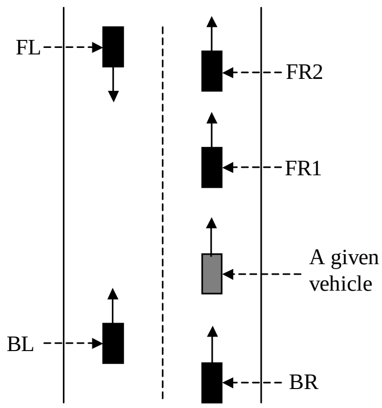


Fig. 4. A given vehicle and five vehicles next to it

During the simulation the system registers a series of parameters of the local simulations, that is simulations connected with each vehicle separately, as well as two global parameters of the simulation that is parameters connected with driving conditions during the simulation. The value of each simulation parameter may vary and what follows it has to be treated as a certain attribute taking values from a specified value set.

We associate the simulation parameters with the readouts of different measuring devices or technical equipment placed inside the vehicle or in the outside environment (e.g., by the road, in a helicopter observing the situation on the road, in a police car). These are devices and equipment playing the role of detecting devices or converters meaning sensors (e.g., a thermometer, range finder, video camera, radar, image and sound converter). The attributes taking the simulation parameter values, by analogy to devices providing their values will be called sensors.

The exemplary sensors are the following: initial and current road (four roads), distance from the crossroads (in screen units), current lane (two lanes), position of the vehicle on the road (values from 0.0 to 1.0), vehicle speed (values from 0.0 to 10.0), acceleration and deceleration, distance of a given vehicle from FR1, FL, BR and BL vehicles and between FR1 and FR2 (in screen units), appearance of the vehicle at the crossroad (binary values), visibility (expressed in screen units values from 50 to 500), humidity (slipperiness) of the road (three values: lack of humidity - dry road, low humidity, high humidity).

If, for some reason, the value of one of the sensors may not be determined, the value of the parameter becomes equal NULL (missing value).

Apart from sensors the simulator registers a few more attributes, whose values are determined using the sensor's values in a way determined by an expert. These parameters in the present simulator version take the binary values and are therefore called concepts. The results returned by testing concepts are very often in a form YES, NO or DOES NOT CONCERN (NULL value).

Here are exemplary concepts:

1. Is the vehicle forcing the right of way at the crossroads?
2. Is there free space on the right lane in order to end the overtaking maneuver?
3. Will the vehicle be able to easily overtake before the oncoming car?
4. Will the vehicle be able to brake before the crossroads?
5. Is the distance from the FR1 vehicle too short or do we predict it may happen shortly?
6. Is the vehicle overtaking safely?
7. Is the vehicle driving safely?

Besides binary concepts, simulator registers for any such concept one special attribute that approximates binary concept by six linearly ordered layers: *certainly YES*, *rather YES*, *possibly YES*, *possibly NO*, *rather NO* and *certainly NO*.

Some concepts related to the situation of the road are simple and classifiers for them can be induced directly from sensor measurement but for more complex concepts this is infeasible. In searching for classifiers for such concepts domain knowledge can be helpful. The relationships between concepts represented in domain knowledge can be used to construct hierarchical relationship diagrams. Such diagrams can be used to induce multi-layered classifiers for complex concepts (see [14] and next section). In Figure 5 there is an exemplary relationship diagram for the above mentioned concepts.

The concept specification and concept dependencies are usually not given automatically in accumulated data sets. Therefore they should be extracted

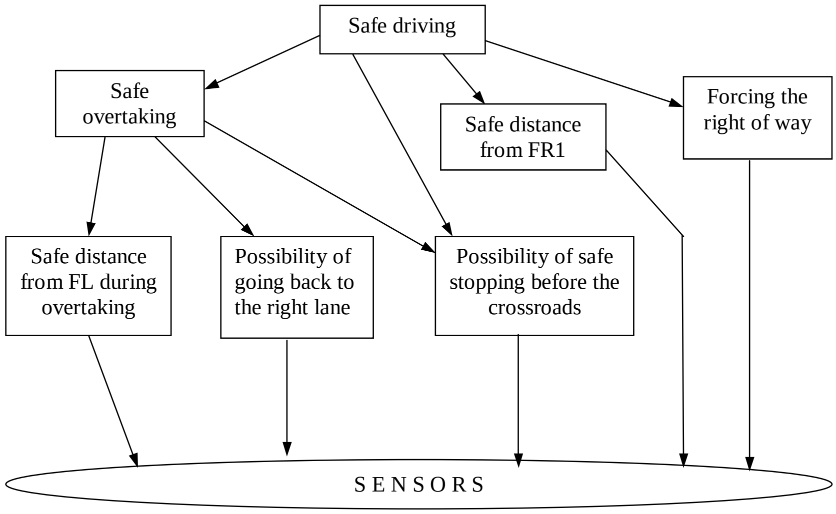


Fig. 5. The relationship diagram for presented concepts

from a domain knowledge. Hence, the role of human experts is very important in our approach.

During the simulation, when a new vehicle appears on the board, its so called driver's profile is determined. It may take one of the following values: a very careful driver, a careful driver and a careless driver. Driver's profile is the identity of the driver and according to this identity further decisions as to the way of driving are made.

Depending on the driver's profile and weather conditions (humidity of the road and visibility) speed limits are determined, which cannot be exceeded.

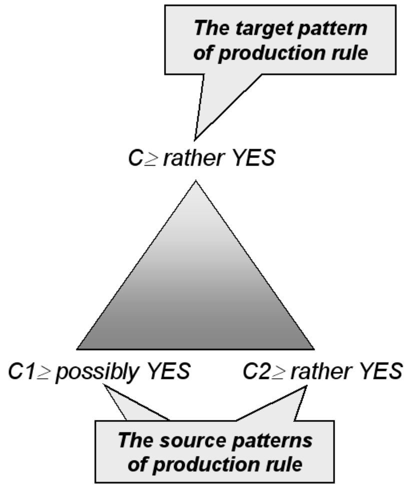
The generated data during the simulation are stored in a data table (information system). Each row of the table depicts the situation of a single vehicle and the sensors' and concepts' values are registered for a given vehicle and the FR1, FR2, FL, BL and BR vehicles (associated with a given vehicle). Within each simulation step descriptions of situations of all the vehicles on the road are saved to file.

4 Algorithm for classifying objects by production

In this section we present an algorithm for classifying objects by a given production but first of all we have to describe the method for the production inducing.

To outline a method for production inducing let us assume that a given concept *C* registered by road simulator depends on two concepts *C1* and *C2* (registered be road simulator too). Each of these concepts can be approximated by six linearly ordered layers: *certainly YES*, *rather YES*, *possibly YES*,

C1	C2	C
certainly YES	*certainly YES*	***certainly YES***
certainly NO	*certainly NO*	*certainly NO*
rather YES	*certainly YES*	***rather YES***
possibly YES	*possibly NO*	*possibly YES*
possibly YES	*possibly NO*	*rather NO*
possibly YES	***rather YES***	***rather YES***
possibly YES	*certainly NO*	*possibly NO*
certainly YES	***rather YES***	***certainly YES***
certainly NO	*possibly NO*	*certainly NO*



certainly YES* > *rather YES* > *possibly YES* > *possibly NO* > *rather NO* > *certainly NO

Fig. 6. The illustration of production rule extracting

possibly NO, *rather NO* and *certainly NO*. We induce classifiers for concepts *C1* and *C2*. These classifiers generate the corresponding weight (the name of one of six approximation layers) for any tested object. We construct for the target concept C a table T over the Cartesian product of sets defined by weight patterns for *C1*, *C2*, assuming that some additional constraints hold. Next, we add to the table T the last column, that is an expert decision. From the table T, we extract production rules describing dependencies between these three concepts.

In Figure 6 we illustrate the process of extracting production rule for concept C and for the approximation layer *rather YES* of concept C. The production rule can be extracted in the following four steps:

1. Select all rows from the table T in which values of column C is not less than *rather YES*.
2. Find minimal values of attributes *C1* and *C2* from table T for selected rows in the previous step (in our example it easy to see that for the attribute *C1* minimal value is *possibly YES* and for the attribute *C2* minimal value is *rather YES*).
3. Set sources patterns of new production rule on the basis of minimal values of attributes that were found in the previous step.
4. Set the target pattern of new production, i.e., concept C with the value *rather YES*.

Finally, we obtain the production rule:
(*) If (*C1* $\geq$ *possibly YES*) and (*C2* $\geq$ *rather YES*) then ($C \geq$ *rather YES*).

A given tested object can be classified by the production rule (*), when weights generated for the object by classifiers induced for concepts from the rule premise are at least equal to degrees from source (premise) patterns of the rule. Then the production rule classifies tested object to the target (conclusion) pattern.

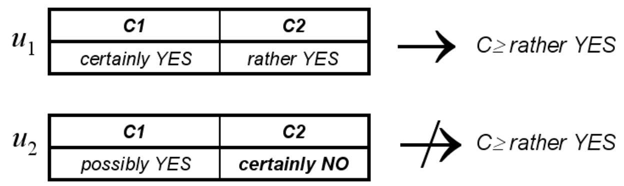


Fig. 7. Classifying tested objects by single production rule

For example, the object u_1 from Figure 7 is classified by production rule (*) because it is matched by the both patterns from the left hand side of the production rule (*) whereas, the object u_2 from figure 7 is not classified by production rule (*) because it is not matched by the second source pattern of production rule (*) (the value of attribute *C2* is less than *rather YES*).

The method of extracting production rule presented above can be applied for various values of attribute *C*. In this way, we obtain a collection of production rules, that we mean as a production.

Using production rules selected from production we can compose AR schemes (see Section 2). In this way relevant patterns for more complex concepts are constructed. Any tested object is classified by AR scheme, if it is matched by all sensory patterns from this AR scheme.

The method of object classification based on production can be described as follows:

1. Preclassify object to the production domain.
2. Classify object by production.

We assume that for any production a production guard (boolean expression) is given. Such a guard describes the production domain and is used in preclassification of tested objects. The production guard is constructed using domain knowledge. An object can be classified by a given production if it satisfies the production guard. For example, let us assume that the production P is generated for the concept: "Is the vehicle overtaking safely ?". Then an object-vehicle u is classified by production P iff u is overtaking. Otherwise, it is returned a message *"HAS NOTHING TO DO WITH (OVERTAKING)"*.

Now, we can present algorithm for classifying objects by production.

Algorithm 1. *The algorithm for classifying objects by production*
Step 1 Select a complex concept C from relationship diagram.
Step 2 If the tested object should not be classified by a given production P

extracted for the selected concept C, i.e., it does not satisfy
the production guard: **return** *HAS NOTHING TO DO WITH*

Step 3 Find a rule from production P that classifies object with
the maximal degree to the target concept of rule
if such a rule of P does not exist **return** *I DO NOT KNOW*.

Step 4 Generate a decision value for object from the degree extracted in
the previous step
if (the extracted degree is greater than *possibly YES*) then
the object is classified to the concept C (**return** *YES*)
else
the object is not classified to the concept C (**return** *NO*).

The algorithm for classifying objects by production presented above can be treated as an algorithm of dynamical synthesis of AR scheme for any tested object. It is easy to see, that during classification any tested object is classified by single production rule selected from production. It means that the production rule is dynamically assigned to the tested object. In other words, the approximating reasoning scheme is dynamically synthesized for any tested object.

We claim that the quality of the classifier presented above is better than the classifier constructed using algorithm based on the set of minimal decision rules. In the next section we present the results of experiments with data sets generated by road simulator supporting this claim.

5 Experiments with Data

To verify effectiveness of classifiers based on AR schemes, we have implemented our algorithms in the AS-lib programming library. This is an extension of the RSES-lib 2.1 programming library creating the computational kernel of the RSES system [16].

The experiments have been performed on the data sets obtained from the road simulator. We have applied the train and test method for estimating accuracy (see e.g. [5]). Data set consists of 18101 objects generated by the road simulator. This set was randomly divided to the train table (9050 objects) and the test table (9051 objects).

In our experiments, we compared the quality of two classifiers: RS and ARS. For inducing of RS we use RSES system generating the set of minimal decision rules that are next used for classifying situations from testing data. ARS is based on AR schemes. We compared RS and ARS classifiers using accuracy of classification, learning time and the rule set size. We also checked the robustness of classifiers.

Table 1 and table 2 show the results of the considered classification algorithms for the concept: “Is the vehicle overtaking safely?” and for the concept “Is the vehicle driving safely?” respectively.

Table 1. Results of experiments for the concept: "Is the vehicle overtaking safely?"

Decision class	Method	Accuracy	Coverage	Real accuracy
YES	RS	0.949	0.826	0.784
	ARS	0.973	0.974	0.948
NO	RS	0.979	0.889	0.870
	ARS	0.926	1.0	0.926
All classes	RS	0.999	0.996	0.995
(YES + NO)	ARS	0.999	0.999	0.998

Table 2. Results of experiments for the concept: "Is the vehicle driving safely?"

Decision class	Method	Accuracy	Coverage	Real accuracy
YES	RS	0.978	0.946	0.925
	ARS	0.962	0.992	0.954
NO	RS	0.633	0.740	0.468
	ARS	0.862	0.890	0.767
All classes	RS	0.964	0.935	0.901
(YES + NO)	ARS	0.958	0.987	0.945

Table 3. Learning time and the rule set size for concept: "Is the vehicle driving safely?"

Method	Learning time	Rule set size
RS	801 seconds	835
ARS	247 seconds	189

One can see that accuracy of algorithm ARS is higher than the accuracy of the algorithm RS for analyzed data set.

Table 3 shows the learning time and the number of decision rules induced for the considered classifiers. In case of the number of decision rules we present the average over all concepts (from the relationship diagram) number of rules.

One can see that the learning time for ARS is much shorter than for RS and the number of decision rules induced for ARS is much lower than the number of decision rules induced for RS.

6 Summary

We have discussed a method for construction (from data and domain knowledge) of classifiers for complex concepts using AR schemes (ARS classifiers).

The experiments showed that:

- the accuracy of classification by ARS is better than accuracy of RS classifier,
- the learning time for ARS is much shorter than for RS,

- the number of decision rules induced for ARS is much lower than the number of decision rules induced for RS.

Finally, the ARS classifier is much more robust than the RS classifier. The results are consistent with the rough-mereological approach.

Acknowledgement. The research has been supported by the grant 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

References

1. Bazan J. (1998) A comparison of dynamic non-dynamic rough set methods for extracting laws from decision tables. In: [8]: 321–365
2. Bazan J., Nguyen H. S., Skowron A., Szczuka M. (2003) A view on rough set concept approximation. LNAI 2639, Springer, Heidelberg: 181–188
3. Friedman J. H., Hastie T., Tibshirani R. (2001) The elements of statistical learning: Data mining, inference, and prediction. Springer, Heidelberg.
4. Kloesgen W., Żytkow J. (eds) (2002) Handbook of KDD. Oxford University Press
5. Michie D., Spiegelhalter D.J., Taylor, C.C. (1994) Machine learning, neural and statistical classification. Ellis Horwood, New York
6. Pawlak Z (1991) Rough sets: Theoretical aspects of reasoning about data. Kluwer, Dordrecht.
7. Pal S. K., Polkowski L., Skowron A. (eds) (2004) Rough-Neuro Computing: Techniques for Computing with Words. Springer-Verlag, Berlin.
8. Polkowski L., Skowron A. (eds) (1998) Rough Sets in Knowledge Discovery 1-2, Physica-Verlag, Heidelberg.
9. Polkowski L., Skowron, A. (1999) Towards adaptive calculus of granules. In: [17]: 201–227
10. Polkowski L., Skowron A. (2000) Rough mereology in information systems. A case study: Qualitative spatial reasoning. In: Polkowski L., Lin T. Y., Tsumoto S. (eds), Rough Sets: New Developmentsin Knowledge Discovery in Information Systems, Studies in Fuzziness and Soft Computing 56, Physica-Verlag, Heidelberg: 89–135
11. Skowron, A. (2001) Toward intelligent systems: Calculi of information granules. Bulletin of the International Rough Set Society 5 (1-2): 9–30
12. Skowron A., Stepaniuk J. (2001) Information granules: Towards foundations of granular computing. International Journal of Intelligent Systems 16 (1): 57–86
13. Skowron A., Stepaniuk J. (2002) Information granules and rough-neuro computing. In [7]: 43–84
14. Stone P. (2000) Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. The MIT Press, Cambridge, MA
15. The Road simulator Homepage – `logic.mimuw.edu.pl/∼bazan/simulator`
16. The RSES Homepage – `logic.mimuw.edu.pl/∼rses`
17. Zadeh L. A., Kacprzyk J. (eds) (1999) Computing with Words in Information/Intelligent Systems 1-2. Physica-Verlag, Heidelberg