Rough Set Methods in Approximation of Hierarchical Concepts

Jan G. Bazan[1], Sinh Hoa Nguyen[2], Hung Son Nguyen[3], and Andrzej Skowron[3]

[1] Institute of Mathematics,University of Rzeszów
Rejtana 16A, 35-959 Rzeszów, Poland
[2] Japanese-Polish Institute of Information Technology
Koszykowa 86, 02-008, Warsaw, Poland
[3] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
{bazan,hoa,son,skowron}@mimuw.edu.pl

Abstract. Many learning methods ignore domain knowledge in synthesis of concept approximation. We propose to use hierarchical schemes for learning approximations of complex concepts from experimental data using inference diagrams based on domain knowledge. Our solution is based on the rough set and rough mereological approaches. The effectiveness of the proposed approach is performed and evaluated on artificial data sets generated by a traffic road simulator.

1 Introduction

Many problems in machine learning, pattern recognition or data mining can be formulated as searching related to concept approximation [5]. A typical concept approximation process uses a given information about objects from a finite subset of universe, called training set or sample, to induce the description of the approximation. In many learning tasks, e.g., identification of dangerous situations on the road by unmanned vehicle aircraft (UAV), the target concept is too complex and it can not be approximated directly from feature value vectors. If the target concept is a composition of some simpler ones, the layered learning [17] is an alternative approach to concept approximation.

Assuming that a hierarchical concept decomposition is given, the main idea is to gradually synthesize a target concept from simpler ones. A learning process can be imagined as a treelike structure with the target concept located at the highest layer. At the lowest layer, basic concepts are approximated using feature values available from a data set. At the next layer more complex concepts are synthesized from the basic concepts. This process is repeated for successive layers.

The importance of hierarchical concept synthesis is now well recognized by researchers (see, e.g., [8, 11, 12]). An idea of hierarchical concept synthesis, in the rough mereological and granular computing frameworks has been developed (see, e.g., [8, 12, 13]) and problems of compound concept approximation are discussed, e.g., in [3, 8, 14, 16].

S. Tsumoto et al. (Eds.): RSCTC 2004, LNAI 3066, pp. 346–355, 2004.

In this paper we deal with concepts that are specified by decision classes in decision systems [9]. The crucial for inducing concept approximations is to create the description of concepts in such a way that makes it possible to maintain the acceptable level of imprecision along all the way from the basic attributes to the final decision. We discuss some strategies for concept composing founded on the rough set approach. We also examine effectiveness of layered learning approach by comparison with the standard rule-based learning approach. Quality of the new approach is verified with respect to the robustness of concept approximation, preciseness of concept approximation, computation time required for concept induction, and concept description length. Experiments are carried out on artificial data sets generated by a traffic road simulator.

2 Rough Set Approach to Concept Approximation

Formally, the concept approximation problem can be formulated as follows: given an universe $\mathcal{U}$ of objects (cases, states, patients, observations, etc.), and a concept X which can be interpreted as a subset of $\mathcal{U}$, the problem is to find a description of X, that can be expressed in a predefined descriptive language $\mathcal{L}$. We assume that $\mathcal{L}$ consists of such formulas that are interpretable as subsets of $\mathcal{U}$.

There are many reasons that force as to find some approximated rather than exact description of a given concept. Let us recall some of them: (i) not satisfactory expressive power of language $\mathcal{L}$ in the universe $\mathcal{U}$: in many learning tasks, the concept X is already defined in some language $\mathcal{L}^*$, (e.g., natural language), but we have to describe X in another, usually poorer language $\mathcal{L}$ (e.g., consisting of boolean formulae defined by some features); (ii) a given concept X is specified partially: in inductive learning approach, values of characteristic function of X are given only for objects from a *training set* $U \subseteq \mathcal{U}$ of objects.

Rough set theory offers an interesting idea to describe a concept in such situations. In the following section, we recall the rough set approach to concept approximation problem. Let us fix some notation used in next sections.

Usually, we assume that the input data for concept approximation problem is given by a decision table, i.e., a tuple $\mathbb{S} = (U, A, dec)$, where U is a non-empty, finite set of *training objects*, A is a non-empty, finite set, of *attributes* and $dec \notin A$ is a distinguished attribute called *decision*. Each attribute $a \in A$ is a function $a : U \to V_a$ called *evaluation function*, where V_a is called the *domain* of a. For any non-empty set of attributes $B \subseteq A$ and any object $x \in U$, we define the *B-signature* of x by: $inf_B(x) = \{(a, a(x)) : a \in B\}$. The set $INF_B(\mathbb{S}) = \{inf_B(x) : x \in U\}$ is called the *B-signature* of $\mathbb{S}$.

Without loss of generality, we assume that the domain of the decision dec is equal to $V_{dec} = \{1, \ldots, d\}$. For any $k \in V_{dec}$, the set $CLASS_k = \{x \in U : dec(x) = k\}$ is called the k^{th} *decision class of* $\mathbb{S}$. The decision dec determines a partition of U into decision classes, i.e., $U = CLASS_1 \cup \ldots \cup CLASS_d$. Rough set methodology for concept approximation can be described as follows.

Definition 1. *Let $X \subseteq \mathcal{U}$ be a concept and let $U \subseteq \mathcal{U}$ be a finite sample of $\mathcal{U}$. Assume that for any $x \in U$ there is given information if $x \in X \cap U$ or*

$x \in U - X$. An approximation (induced from sample U) of the concept X is any pair $\mathbb{P} = (\mathbf{L}, \mathbf{U})$ satisfying the following conditions:

1. $\mathbf{L} \subseteq \mathbf{U} \subseteq \mathcal{U}$;
2. $\mathbf{L}, \mathbf{U}$ *are subsets of* $\mathcal{U}$ *expressible in the language* $\mathcal{L}$;
3. $\mathbf{L} \cap U \subseteq X \cap U \subseteq \mathbf{U} \cap U$;
4. *the set* $\mathbf{L}$ *(*$\mathbf{U}$*) is maximal (minimal) in the family of sets definable in* $\mathcal{L}$ *satisfying (3).*

The sets $\mathbf{L}$ and $\mathbf{U}$ are called the *lower approximation* and the *upper approximation* of the concept $X \subseteq \mathcal{U}$, respectively. The set $\mathbf{BN} = \mathbf{U} \setminus \mathbf{L}$ is called the *boundary region of approximation* of X. The set X is called *rough* with respect to its approximations $(\mathbf{L}, \mathbf{U})$ if $\mathbf{L} \neq \mathbf{U}$, otherwise X is called *crisp* in $\mathcal{U}$. The pair $(\mathbf{L}, \mathbf{U})$ is also called the *rough set* (for the concept X). The condition (4) in the above list can be substituted by inclusion to a degree to make it possible to induce approximations of higher quality of the concept on the whole universe $\mathcal{U}$. In practical applications the last condition in the above definition can be hard to satisfy. Hence, by using some heuristics we construct sub-optimal instead of maximal or minimal sets. The rough approximation of concept can be also defined by means of rough membership function.

Definition 2. *Let $X \subseteq \mathcal{U}$ be a concept and let decision table $\mathbb{S} = (U, A, dec)$ describe the set of training objects $U \subseteq \mathcal{U}$. A function $f : \mathcal{U} \to [0, 1]$ is called a rough membership function of the concept $X \subseteq \mathcal{U}$ if, and only if $(\mathbf{L}_f, \mathbf{U}_f)$ is a rough approximation of X (induced from sample U), where $\mathbf{L}_f = \{x \in \mathcal{U} : f(x) = 1\}$ and $\mathbf{U}_f = \{x \in \mathcal{U} : f(x) > 0\}$.*

Many methods of discovering rough approximations of concepts from data have been proposed, e.g., method based on reducts [9][10], on k-NN classifiers [3], or on decision rules [3]. In the next section we will use rough membership functions to construct the layered learning algorithm. Hence, let us recall now a construction of rough membership function for concept approximation. The construction is based on decision rules.

Searching for decision rules, which are *short, strong* and *having high confidence*, from a given decision table is a challenge for data mining. Many methods based on rough set theory have been proposed to deal with such problems (see, e.g., [2, 6]).

Given a decision table $\mathbb{S} = (U, A, dec)$. Let us assume that $\mathbf{RULES}(\mathbb{S})$ is a set of decision rules induced by some rule extraction method. For any object $x \in \mathcal{U}$, let $MatchRules(\mathbb{S}, x)$ be the set of rules from $\mathbf{RULES}(\mathbb{S})$ supported by x. One can define the rough membership function $\mu_{CLASS_k} : \mathcal{U} \to [0, 1]$ for the concept determined by $CLASS_k$ as follows:

1. Let $\mathbf{R}_{yes}$ be the set of all decision rules from $MatchRules(\mathbb{S}, x)$ for k^{th} class and let $\mathbf{R}_{no} \subset MatchRules(\mathbb{S}, x)$ be the set of decision rules for other classes.
2. We define two real values w_{yes}, w_{no}, called "for" and "against" weights for the object x by

$$w_{yes} = \sum_{\mathbf{r} \in \mathbf{R}_{yes}} strength(\mathbf{r}) \qquad w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}} strength(\mathbf{r}) \tag{1}$$

where $strength(\mathbf{r})$ is a normalized function depending on $length$, $support$, $confidence$ of $\mathbf{r}$ and some global information about the decision table $\mathbb{S}$ like table size, class distribution (see [2]).

3. One can define the value of $\mu_{CLASS_k}(x)$ by

$$\mu_{CLASS_k}(x) = \begin{cases} \text{undetermined} & \text{if } \max(w_{yes}, w_{no}) < \omega \\ 0 & \text{if } w_{no} - w_{yes} \geq \theta \text{ and } w_{no} > \omega \\ 1 & \text{if } w_{yes} - w_{no} \geq \theta \text{ and } w_{yes} > \omega \\ \frac{\theta + (w_{yes} - w_{no})}{2\theta} & \text{in other cases} \end{cases}$$

where ω, θ are parameters set by user. These parameters make it possible in a flexible way to control the size of boundary region for the approximations established according to Definition 2.

3 Layered Learning Approach Based on Rough Set Theory

In this section we discuss a composing strategy for concepts from already existing ones. Such strategy realizes a crucial step in concept synthesis. We discuss a method that makes it possible to control the level of approximation quality along all the way from basic concepts to the target concept.

We assume that a concept hierarchy H is given. The concept hierarchy should contain either inference diagram or dependence diagram that connect the target concept with input attributes through intermediate concepts. A training set is represented by decision table $\mathbb{S}_S = (U, A, D)$, where D is a set of decision attributes corresponding to all intermediate concepts and to the target concept. Decision values indicate if an object belong to basic concepts and to the target concept, respectively.

Using information available from a concept hierarchy for each basic concept C_b one can create a training decision system $\mathbb{S}_{C_b} = (U, A_{C_b}, dec_{C_b})$, where $A_{C_b} \subseteq A$, and $dec_{C_b} \in D$. To approximate the concept C_b one can apply any classical method (e.g., k-NN, supervised clustering, or rule-based approach [7]) to the table $\mathbb{S}_{C_b}$. In further discussion we assume that basic concepts are approximated by rule based classifiers (see Section 2) derived from relevant decision tables.

To avoid overly complicated notation let us limit ourselves to the case of constructing compound concept approximation on the basis of two simpler concept approximations. Assume we have two concepts C_1 and C_2 that are given to us in the form of rule-based approximations derived from decision systems $\mathbb{S}_{C_1} = (U, A_{C_1}, dec_{C_1})$ and $\mathbb{S}_{C_1} = (U, A_{C_1}, dec_{C_1})$. Henceforth we are given two rough membership functions $\mu_{C_1}(x)$, $\mu_{C_2}(x)$. These functions are determined with use of parameter sets $\{w_{yes}^{C_1}, w_{no}^{C_1}, \omega^{C_1}, \theta^{C_1}\}$ and $\{w_{yes}^{C_2}, w_{no}^{C_2}, \omega^{C_2}, \theta^{C_2}\}$, respectively. We want to establish similar set of parameters $\{w_{yes}^{C}, w_{no}^{C}, \omega^{C}, \theta^{C}\}$ for the target concept C, which we want to describe with use of rough membership function μ_C. As previously, the parameters ω, θ controlling of the boundary region are user-configurable. But, we need to derive $\{w_{yes}^{C}, w_{no}^{C}\}$ from the data.

The issue is to define a decision system from which rules used to define approximations can be derived. To this end we concentrate on this matter.

We assume that both simpler concepts C_1, C_2 and the target concept C are defined over the same universe of objects $\mathcal{U}$. Moreover, all of them are given on the same sample $U \subset \mathcal{U}$. To complete the construction of the decision system $\mathbb{S}_C = (U, A_C, dec_C)$ we need to specify the conditional attributes from A_C and the decision attribute dec_C. The decision attribute value $dec_C(x)$ is given for any object $x \in U$. For conditional attributes, we assume that they are either rough membership functions for simpler concepts (i.e., $A_C = \{\mu_{C_1}(x), \mu_{C_2}(x)\}$) or weights for simpler concepts (i.e., $A_C = \{w_{yes}^{C_1}, w_{no}^{C_1}, w_{yes}^{C_2}, w_{no}^{C_2}\}$). The output set O_i for each concept C_i, where $i = 1, 2$, consists of either one attribute which is a rough membership function μ_{C_i} (in the first case) or two attributes $w_{yes}^{C_i}, w_{no}^{C_i}$ which describe fitting degrees of objects to the concept C_i and its complement, respectively.

By extracting rules from $\mathbb{S}_C$ rule-based approximations of the concept C are created. Algorithm 1 is the layered learning algorithm used in our experiments.

Algorithm 1 Layered learning algorithm

Input: Decision system $\mathbb{S} = (U, A, d)$, concept hierarchy H;
Output: Schema for concept composition

for $l := 0$ to max_level **do**
  for (any concept C_k at the level l in H) **do**
    if $l = 0$ **then**
      $\mathbb{S}_{C_k} := (U, A_k, dec_{C_k})$; $\{A_k \subset A$ is a set of attributes relevant to $C_k\}$
    else
      $A_k := \bigcup O_{k_i}$; {where the sum is taken for all sub-concepts C_{k_i} of C_k}
      $\mathbb{S}_{C_k} := (U, A_k, dec_{C_k})$;
    end if
    generate a rule set determining of the concept C_k approximation;
    generate the output vector $O_k = \{w_{yes}^{C_k}, w_{no}^{C_k}\}$, where $w_{yes}^{C_k}(x)$ and $w_{no}^{C_k}(x)$ are the fitting degree of object x to the concept C_k and its complement.
  end for
end for

It is important to observe that such rules describing C use attributes that are in fact classifiers themselves. Therefore, in order to have more readable and intuitively understandable description as well as more control over quality of approximation (especially for new cases) it pays to stratify and interpret attribute domains for attributes in A_C. Instead of using just a value of membership function or weight we would prefer to use linguistic statements such as *"the likeliness of the occurrence of* C_1 *is low"*. In order to do that we have to map the attribute value sets onto some limited family of subsets. Such subsets are then identified with notions such us *"certain"*, *"low"*, *"high"* etc. It is quite natural, especially in case of attributes being membership functions, to introduce linearly ordered subsets of attribute ranges, e.g., $\{negative, low, medium, high, positive\}$. That yields fuzzy-like layout, or linguistic variables, of attribute values. One may (and in some cases should) consider also the case when these subsets overlap.

Stratification of attribute values and introduction of linguistic variable attached to inference hierarchy serves multiple purposes. First, it provides a way for representing knowledge in more human-readable format since if we have a new situation (new object $x^* \in \mathcal{U} \setminus U$) to be classified (checked against compliance with concept C) we may use rules like: ***If*** *compliance of* x^* *with* C_1 *is* *high or medium* ***and*** *compliance of* x^* *with* C_2 *is* *high* ***then*** $x^* \in C$.

Another advantage of imposing the division of attribute value sets lays in extended control over flexibility and validity of system constructed in this way. As we may define the linguistic variables and corresponding intervals, we gain the ability of making system more stable and inductively correct. In this way we control the general layout of boundary regions for simpler concepts that contribute to construction of the target concept. The process of setting the intervals for attribute values may be performed by hand, especially when additional background information about the nature of the described problem is available. One may also rely on some automated methods for such interval construction, such as, e.g., clustering, template analysis and discretization. Some extended discussion on foundations of this approach, which is related to rough-neural computing [8] and computing with words can be found in [15, 16].

4 Experimental Results

To verify a quality of hierarchical classifiers we performed some experiments with the road simulator system.

4.1 Road Simulator

Learning to recognize and predict traffic situations on the road is the main issue in many unmanned vehicle aircraft (UVA) projects. It is a good example for the hierarchical concept approximation problem. We demonstrate the proposed layered learning approach on our own simulation system.

ROAD SIMULATOR is a computer tool generating data sets consisting of recording vehicle movements on the roads and at the crossroads. Such data sets are next used to learn and test complex concept classifiers working on information coming from different devices (sensors) monitoring the situation on the road. Let us present some important features of this system.

During the simulation the system registers a series of parameters of the local simulations, that is simulations related to each vehicle separately, as well as two global parameters of the simulation that is parameters specifying driving conditions during the simulation. The local parameters are related to driver's profile, which is randomly determined, when a new vehicle appears on the board, and may not be changed until it disappears from the board. The global parameters like visibility, weather conditions are set randomly according to some scenario. 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 police car, etc.). Apart from those sensors, the simulator registers a few more attributes, whose values are determined by the sensor's values in a way specified by an expert. In Figure 1 we present an example of a hierarchical diagram for the some exemplary concepts. During the simulation data may be generated and stored in a text file in a form of a rectangle table (information system). Each row of this table depicts the situation of a single vehicle, e.g., the values of sensors and concepts values. Within each simulation step descriptions of situations of all the vehicles are stored into a file.

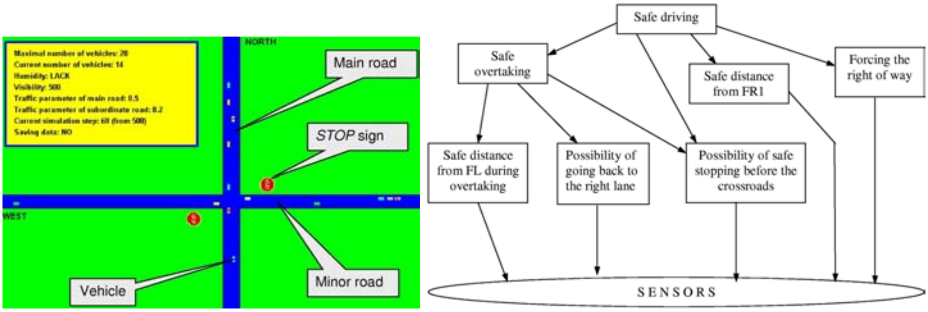

Fig. 1. The board of simulation and the relationship diagram of exemplary concepts

4.2 Experiment Setup

We have generated 6 training data sets: *c10_s100, c10_s200, c10_s300, c10_s400, c10_s500, c20_s500* and 6 corresponding testing data sets named by *c10_s100N, c10_s200N, c10_s300N, c10_s400N, c10_s500N, c20_s500N*. All data sets consists of 100 attributes. The smallest data set consists of above 700 situations (100 simulation units) and the largest data set consists of above 8000 situations (500 simulation units).

We compare the accuracy of two classifiers, i.e., **RS:** the standard classifier induced by the rule set method, and **RS-L:** the hierarchical classifier induced by the RS-layered learning method. In the first approach, we employed the RSES system [4] to generate the set of minimal decision rules. We use the simple voting strategy for conflict resolution in new situation classification.

In the RS-layered learning approach, from training table we create five subtables to learn five basic concepts (see Figure 1): C_1: "*safe distance from FL during overtaking*," C_2: "*possibility of safe stopping before crossroads*," C_3: "*possibility of going back to the right lane*," C_4: "*safe distance from FR1*," C_5: "*forcing the right of way*." A concept C_6: "*safe_overtaking*" is located in the next level. To approximate concept C_6, we create a table with three conditional attributes. These attributes describe fitting degrees of objects to concepts C_1, C_2, C_3, respectively. The target concept C_7: "*safe_driving*" is located in the third level of the concept decomposition hierarchy. To approximate C_7 we also create a decision table with three attributes, representing fitting degrees of objects to the concepts C_4, C_5, C_6, respectively.

Classification Accuracy: Similarly to real life situations, the decision class "safe_driving = YES" is dominating. The decision class "safe_driving = NO" takes only 4% - 9% of training sets. Searching for approximation of "safe_driving = NO" class with the high precision and generality is a challenge of leaning algorithms. Let us concentrate on approximation quality of the "NO" class. In Table 1 we present the classification accuracy of RS and RS-L classifiers. One can observe, the accuracy of "YES" class of both standard and hierarchical classifiers is high. Whereas accuracy of "NO" class is very poor, particularly in case of the standard classifier. The hierarchical classifier showed to be much better than the standard classifier for this class. Accuracy of "NO" class of the hierarchical classifier is quite high when the size of training sets is sufficiently large.

Table 1. Classification accuracy of standard and hierarchical classifiers

Accuracy	**Total**		**Class YES**		**Class of NO**	
	RS	**RS-L**	**RS**	**RS-L**	**RS**	**RS-L**
$c10\_s100N$	0.94	0.97	1	1	0	0
$c10\_s200N$	0.99	0.96	1	0.98	0.75	0.60
$c10\_s300N$	0.99	0.98	1	0.98	0	0.78
$c10\_s400N$	0.96	0.77	0.96	0.77	0.57	0.64
$c10\_s500N$	0.96	0.89	0.99	0.90	0.30	0.80
$c20\_s500N$	0.99	0.89	0.99	0.88	0.44	0.93
Average	0.97	0.91	0.99	0.92	**0.34**	**0.63**

Robustness and Coverage Rate: Robustness and coverage rate of classifiers are evaluated by their recognition ability for unseen situations. The recognition rate of situations belonging to "NO" class is very poor in the case of the standard classifier. One can see in Table 2 the improvement on coverage degree of "YES" class and "NO" class of the hierarchical classifier.

Table 2. Coverage rate of standard and hierarchical classifiers

Covering rate	**Total**		**Class YES**		**Class NO**	
	RS	**RS-L**	**RS**	**RS-L**	**RS**	**RS-L**
$c10\_s100N$	0.44	0.72	0.44	0.74	0.50	0.38
$c10\_s200N$	0.72	0.73	0.73	0.74	0.50	0.63
$c10\_s300N$	0.47	0.68	0.49	0.69	0.10	0.44
$c10\_s400N$	0.74	0.90	0.76	0.93	0.23	0.35
$c10\_s500N$	0.72	0.86	0.74	0.88	0.40	0.69
$c20\_s500N$	0.62	0.89	0.65	0.89	0.17	0.86
Average	0.62	0.79	0.64	0.81	**0.32**	**0.55**

Computing Speed: With respect to time the layered learning approach shows a tremendous advantage in comparison with the standard learning approach. In the case of the standard classifier, computational time is measured as a time required for computing a rule set used to decision class approximation. In the case of hierarchical classifier computational time is the total time required for

Table 3. Time for standard and hierarchical classifier generation (experiments were performed on computer with processor AMD Athlon 1.4GHz., 256MB RAM)

Table names	**RS**	**RS-L**	**Speed up ratio**
$c10\_s100$	94 s	2.3 s	40
$c10\_s200$	714 s	6.7 s	106
$c10\_s300$	1450 s	10.6 s	136
$c10\_s400$	2103 s	34.4 s	60
$c10\_s500$	3586 s	38.9 s	92
$c20\_s500$	10209 s	98.0s	104
Average			**90**

all sub-concepts and target concept approximation. One can see in Table 3 that the speed up ratio of the layered learning approach to the standard one is from 40 to 130.

5 Conclusions

We presented a new method for concept synthesis. It is based on the layered learning approach. Unlike traditional approach, in the layered learning approach the concept approximations are induced not only from accessed data sets but also from expert's domain knowledge. In the paper, we assume that knowledge is represented by concept dependency hierarchy. The layered learning approach showed to be promising for the complex concept synthesis. Experimental results with road traffic simulation are showing advantages of this new approach in comparison to the standard approach. The concept approximation by composition of sub-concepts is the main problem in the layered learning approach. In future we plan to investigate more advanced approaches for concept composition.

One interesting possibility is to use patterns defined by rough approximations of concepts defined by different kinds of classifiers in synthesis of compound concepts. We also would like to develop methods for rough-fuzzy classifier's synthesis (see Section 3). In particular, the mentioned in Section 3 method based on rough-fuzzy classifiers introduces more flexibility for such composing because a richer class of patterns introduced by different layers of rough-fuzzy classifiers can lead to improving of classifier quality [8].

We also plan to apply layered learning approach to real-life problems, especially when domain knowledge is specified in natural language. This can make further links with the computing with words paradigm [8, 18]. This is in particular linked with the rough mereological approach (see, e.g., [12]) and with the rough set approach for approximate reasoning in distributed environments [13, 15], in particular with methods of information system composition [1, 15].

Acknowledgements

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

References

1. Barwise, J., Seligman, J., eds.: Information Flow: The Logic of Distributed Systems. Cambridge University Press, Cambridge, UK (1997)
2. Bazan, J.G.: A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In Polkowski, L., Skowron, A., eds.: Rough Sets in Knowledge Discovery 1: Methodology and Applications. Physica-Verlag, Heidelberg (1998) 321–365
3. Bazan, J., Nguyen, H.S., Skowron, A., Szczuka, M.: A view on rough set concept approximation. LNAI **2639**, Heidelberg, Springer-Verlag (2003) 181–188
4. Bazan, J.G., Szczuka, M.: RSES and RSESlib - a collection of tools for rough set computations. LNAI **2005** , Springer-Verlag, Heidelberg (2001) 106–113
5. Kloesgen, W., Żytkow, J., eds.: Handbook of Knowledge Discovery and Data Mining. Oxford University Press, Oxford (2002)
6. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: a tutorial. In Pal, S.K., Skowron, A., eds.: Rough Fuzzy Hybridization: A New Trend in Decision-Making. Springer-Verlag, Singapore (1999) 3–98
7. Mitchell, T.: Machine Learning. Mc Graw Hill (1998)
8. Pal, S.K., Polkowski, L., Skowron, A., eds.: Rough-Neural Computing: Techniques for Computing with Words. Springer-Verlag, Heidelberg (2003)
9. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Volume 9 of System Theory, Knowledge Engineering and Problem Solving. Kluwer Academic Publishers, Dordrecht (1991)
10. Pawlak, Z., Skowron, A.: A rough set approach for decision rules generation. In: Thirteenth International Joint Conference on Artificial Intelligence IJCAI, Chambéry, France, Morgan Kaufmann (1993) 114–119
11. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Notices of the AMS **50** (2003) 537–544
12. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. International Journal of Approximate Reasoning **15** (1996) 333–365
13. Skowron, A.: Approximate reasoning by agents in distributed environments. In Zhong, N., Liu, J., Ohsuga, S., Bradshaw, J., eds.: Intelligent Agent Technology Research and Development: Proceedings of IAT01, Maebashi, Japan, October 23-26. World Scientific, Singapore (2001) 28–39
14. Skowron, A.: Approximation spaces in rough neurocomputing. In Inuiguchi, M., Tsumoto, S., Hirano, S., eds.: Rough Set Theory and Granular Computing. Springer-Verlag, Heidelberg (2003) 13–22
15. Skowron, A., Stepaniuk, J.: Information granule decomposition. Fundamenta Informaticae **47(3-4)** (2001) 337–350
16. Skowron, A., Szczuka, M.: Approximate reasoning schemes: Classifiers for computing with words. In: Proceedings of SMPS 2002. Advances in Soft Computing, Heidelberg, Canada, Springer-Verlag (2002) 338–345
17. Stone, P.: Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. The MIT Press, Cambridge, MA (2000)
18. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine **22** (2001) 73–84