Improving Rough Classifiers Using Concept Ontology

Nguyen Sinh Hoa[1] and Nguyen Hung Son[2]

[1] Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008, Warsaw, Poland
[2] Institute of Mathematics, Warsaw University,
Banacha 2, 02-097 Warsaw, Poland
{hoa, son}@mimuw.edu.pl

Abstract. We present a method of classifier synthesis based on rough set theory and hierarchical learning idea. The improvement of the generated classifiers is achieved by using concept ontology as a domain knowledge. We examine the effectiveness of the proposed approach by comparing it with standard learning approaches with respect to different criteria. Our experiments are performed on benchmark data set as well as on artificial data sets generated by a road traffic simulator.

1 Introduction

Rough set theory has been introduced by [9] as a tool for concept approximation from uncertainty. Till now, one can find many efficient applications of rough sets in machine learning and data mining, since many problems like classification, clustering or regression can be formulated as concept approximation problem [4]. In a typical process of concept approximation we assume that there is given information consisting of values of conditional and decision attributes on objects from a finite subset (training set) of the universe and using this information one should induce approximations of the concept over the whole universe.

In some 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. The difficulty is based either on the unlearnability of the hypothesis space or on the high complexity of the the learning algorithm. In such cases, there is a need of using a domain knowledge to improve the learning process. In this paper, we assume that domain knowledge is given as a concept ontology, which can be understood as a treelike structure with the target concept located at the root, with attributes (variables, features) located at leaves, and with some additional concepts located in internal nodes. With this assumption, the layered learning [15] seen as a generalization of standard approach to concept approximation.

Given the concept ontology, the main idea is to gradually synthesize a target concept from simpler ones. The importance of hierarchical concept synthesis is now well recognized by researchers (see, e.g., [8] [11]). An idea of hierarchical

T.B. Ho, D. Cheung, and H. Liu (Eds.): PAKDD 2005, LNAI 3518, pp. 312–322, 2005.

concept synthesis, in the rough mereological and granular computing frameworks has been developed (see, e.g., [11] [14]) and problems connected with compound concept approximation are discussed, e.g., in [1] [8] [13].

In this paper we concentrate on 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 basic attributes to final decision. We discuss some strategies for concept composing based on rough set theory. The effectiveness of layered learning approach and the comparison with standard rule-based learning approach are performed with respect to generality of concept approximation, preciseness of concept approximation, computation time required for concept induction and concept description lengths.

2 Basic Notions

The problem of concept approximation can be treated as a problem of searching for description (expressible in a given language) of an unknown concept.

Formally, given an universe $\mathcal{X}$ of objects and a concept C which can be interpreted as a subset of $\mathcal{X}$, the problem is to find a description of C which 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 $\mathcal{X}$. The approximation is required to be as *close* to the original concept as possible.

In this paper, we assume that objects from $\mathcal{X}$ are described by finite set of attributes (features) $A = \{a_1, .., a_k\}$. Each attribute $a \in A$ corresponds to the function $a : \mathcal{X} \to V_a$ where V_a is called the *domain* of a. For any non-empty set of attributes $B \subseteq A$ and any object $x \in \mathcal{X}$, we define the *B-information vector* 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-information set*. The language $\mathcal{L}$, which is used to describe approximations of the given concept, consists of Boolean expressions over descriptors of the form $(attribute = value)$ or $(attribute \in set\_of\_values)$.

Usually, the concept approximation problem is formulated as an *inductive learning problem*, i.e., the problem of searching for a (approximated) description of a concept C based on a *finite set of examples* $U \subset \mathcal{X}$, called the training set. The closeness of the approximation to the original concept can be measured by different criteria like accuracy, description length, etc., which can be also estimated by *test examples*.

The input data for concept approximation problem is given by *decision table* which is 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*. If $C \subset \mathcal{X}$ is a concept to be approximated, then the decision attribute dec is a characteristic function of concept C, i.e., if $x \in C$ we have $dec(x) = yes$, otherwise $dec(x) = no$. In general, the decision attribute dec can describe several disjoint concepts. Therefore, without loss of generality, we assume that the domain of the decision dec is finite and 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., $U = CLASS_1 \cup \ldots \cup CLASS_d$.

The approximated description of a concept can be induced by any learning algorithm from inductive learning area. In the next Section we concentrate on methods based on layered learning and rough set theory.

3 Rough Sets and Concept Approximation Problem

Let $C \subseteq \mathcal{X}$ be a concept and let $\mathbb{S} = (U, A, dec)$ be a decision table describing the training set $U \subseteq \mathcal{X}$. Any pair $\mathbb{P} = (\mathbf{L}, \mathbf{U})$ is called *rough approximation of* C (see [1] [9]) if it satisfies the following conditions:

1. $\mathbf{L} \subseteq \mathbf{U} \subseteq \mathcal{X}$;
2. $\mathbf{L}, \mathbf{U}$ are expressible in the language $\mathcal{L}$;
3. $\mathbf{L} \cap U \subseteq C \cap U \subseteq \mathbf{U} \cap U$;
4. $\mathbf{L}$ is maximal and $\mathbf{U}$ is minimal among those $\mathcal{L}$-definable sets satisfying 3.

The sets $\mathbf{L}$ and $\mathbf{U}$ are called the *lower approximation* and the *upper approximation* of the concept C, respectively. The set $\mathbf{BN} = \mathbf{U} - \mathbf{L}$ is called the *boundary region of approximation* of C. For objects $x \in \mathbf{U}$, we say that "probably, x is in C". The concept C is called *rough* with respect to its approximations $(\mathbf{L}, \mathbf{U})$ if $\mathbf{L} \neq \mathbf{U}$, otherwise C is called *crisp* in $\mathcal{X}$.

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 $\mathcal{X}$. 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.

3.1 Rough Classifier

The rough approximation of a concept can be also defined by means of a rough membership function. A function $\mu_C : \mathcal{X} \to [0, 1]$ is called a rough membership function of the concept $C \subseteq \mathcal{X}$ if, and only if $(\mathbf{L}_{\mu_C}, \mathbf{U}_{\mu_C})$ is a rough approximation of C, where $\mathbf{L}_{\mu_C} = \{x \in \mathcal{X} : \mu_C(x) = 1\}$ and $\mathbf{U}_{\mu_C} = \{x \in \mathcal{X} : \mu_C(x) > 0\}$ (see [1]). The rough membership function can be treated as a fuzzyfication of rough approximation. It makes the translation from rough approximation into membership function. The main feature that stands out rough membership functions is related to the fact that it is derived from data. Any algorithm that computes the value of a rough membership function $\mu_C(x)$ having information vector $inf(x)$ of an object $x \in \mathcal{X}$ as an input, is called *the rough classifier*.

Rough classifiers are constructed from training decision table. Many methods of construction of rough classifiers have been proposed, e.g., the classical method based on reducts [9][10], the method based on k-NN classifiers [1], or the method based on decision rules [1]. Let us remind the Rough Set based algorithm, called *RS algorithm*, that constructs rough classifiers from decision rules. This method will be improved in the next section.

Let $\mathbb{S} = (U, A, dec)$ be a given decision table. The first step of RS algorithm is construction of some decision rules, i.e., implications of a form

$$\mathbf{r} \quad \underset{df}{\equiv} \quad (a_{i_1} = v_1) \wedge ... \wedge (a_{i_m} = v_m) \Rightarrow (dec = k) \tag{1}$$

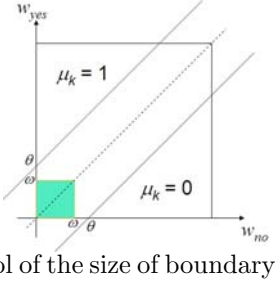
where $a_{i_j} \in A$, $v_j \in V_{a_{i_j}}$ and $k \in V_{dec}$. Searching for *short, strong* decision rules *with high confidence* from a given decision table is a big challenge for data mining. Some methods based on rough set theory have been presented in [3] [5] [10] [12]. Let **RULES**($\mathbb{S}$) be a set of decision rules induced from $\mathbb{S}$ by one of the mentioned rule extraction methods. One can define the rough membership function $\mu_k : \mathcal{X} \to [0, 1]$ for the concept determined by $CLASS_k$ as follows:

1. For any object $x \in \mathcal{X}$, let $MatchRules(\mathbb{S}, x)$ be the set of rules which are supported by x. Let $\mathbf{R}_{yes}$ be the set of all decision rules from $MatchRules(\mathbb{S}, x)$ for k^{th} class and let $\mathbf{R}_{no}$ be the remainder of $\mathbf{R}_{yes}$.
2. We define two real values w_{yes}, w_{no} by

$$w_{yes} = \sum_{\mathbf{r} \in \mathbf{R}_{yes}} strength(\mathbf{r}) \text{ and } w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}} strength(\mathbf{r})$$

   where $strength(\mathbf{r})$ is a normalized function depending on *length*, *support*, $con\!fidence$ of $\mathbf{r}$ and some global information about the decision table $\mathbb{S}$ like table size, class distribution (see [12][1]).
3. The value of $\mu_k(x)$ is defined by:

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



   Parameters ω, θ should be tuned by the user to control of the size of boundary region. They are very important in layered learning approach based on rough set theory.

3.2 Construction of Complex Rough Classifier from Concept Ontology

In this section we describe a strategy that learns to approximate the concept established on the higher level of a given ontology by composing approximations of concepts located at the lower level. We will discuss the method that gives us the ability to control the level of the approximation quality along all the way from attributes (basic concepts) to the target concept.

Let us assume that a concept hierarchy (or a ontology of concepts) is given. The concept hierarchy should contain either inference diagram or dependence

diagram that connects the target concept with input attribute through intermediate concepts. Formally, any concept hierarchy can be treated as a treelike structure $\mathbb{H} = (\mathcal{C}, \mathcal{R})$, where $\mathcal{C}$ is a set of all concepts in the hierarchy including basic concepts (input attributes), intermediated concepts and target concept and $\mathcal{R} \subset \mathcal{C} \times \mathcal{C}$ is a dependency relation between concepts from $\mathcal{C}$. Usually, concept hierarchy is a rooted tree including target concept at root and input attributes at leaves. We also assume that concepts are divided into levels in such a way that every concept is connected with concepts in the lower levels only. Some examples of concept hierarchy are presented in Fig. 2 and Fig. 4.

In Section 3.1, we presented a classical approach (the RS algorithm) to concept approximation problem. This algorithm works for *flat* hierarchy of concepts (i.e., the target concept (decision attribute) is connected directly to input attributes). The specification of RS algorithm is as follows:

Input: Given decision table $\mathbb{S}_C = (U, A_C, dec_C)$ for a flat concept hierarchy (containing C on the top and attributes from A_C on the bottom);
Parameters: ω_C, θ_C;
Output: Approximation of C, i.e., such a set of hypothetical classifiers h_C that indicates the membership of any object x (x not necessary belongs to U) to the concept C. Let us suppose that $h_C(x) = \{\mu_C(x), \mu_{\overline{C}}(x)\}$, where $\overline{C}$ is a complement of the concept C.

For more complicated concept hierarchies, we can use the RS algorithm as a building block to develop a layered learning algorithm. The idea is to apply the RS algorithm to approximate the successive concepts through the hierarchy (from leaves to target concepts). Let $prev(C) = \{C_1, ..., C_m\}$ be the set of concepts in the lower layers, which are connected with C in the hierarchy. The rough approximation of the concept C can be determined by two steps:

1. Construct a decision table $\mathbb{S}_C = (U, A_C, dec_C)$ appropriated the concept C;
2. Apply RS algorithm to extract an approximation of C from $\mathbb{S}_C$;

The main trouble in layered learning algorithms is based on the construction of an adequate decision table. In this paper, we assume that the set of training objects U is common for the whole hierarchy. The set of attributes A_C is strictly related to concepts $C_1, ..., C_m$ in the set $prev(C)$, i.e., $A_C = h_{C_1} \cup h_{C_2} \cup ... \cup h_{C_m}$, where h_{C_i} denotes the set of hypothetical attributes related to the concept C_i. If C_i is an input attribute $a \in A$ then $h_{C_i}(x) = \{a(x)\}$, otherwise $h_C(x) = \{\mu_C(x), \mu_{\overline{C}}(x)\}$. The idea is illustrated in Fig. 1.

The problem which often occurs in layered learning algorithm is related to the lack of decision attributes for intermediate concepts (see Section 4.1). In such situations, we use a supervised clustering algorithm (using decision attribute of the target concept as a class attribute) to create a synthetic decision attribute.

A training set for layered learning is represented by decision table $\mathbb{S}_{\mathbb{H}} = (U, A, D)$, 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 the given concept in the ontology. The most advanced feature of the

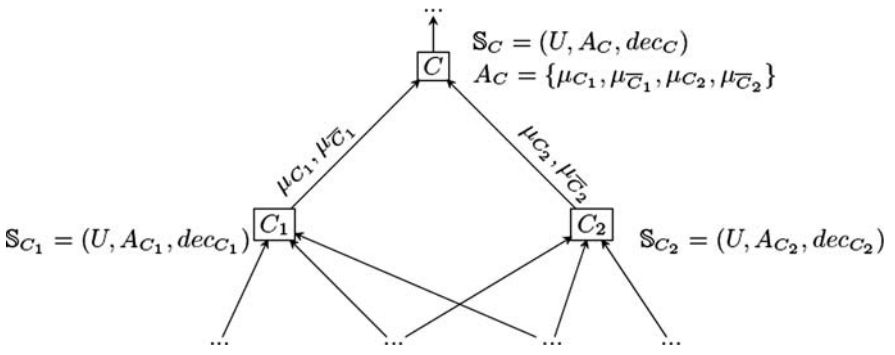


Fig. 1. The construction of decision table for a higher level concept using rough approximation of concepts from lower level

proposed method is the possibility of tuning the quality of concept approximation process via the parameters ω_C, θ_C. More details about this problem will be discussed in our next contribution. The layered learning algorithm based on rough set theory is presented in Algorithm 1.

Algorithm 1. Layered learning algorithm

Input: Decision system $\mathbb{S} = (U, A, D)$, concept hierarchy $\mathbb{H}$;
Output: Hypothetical attributes of all concepts in the hierarchy

```
for l := 0 to max_level do
  for (any concept C_k at the level l in H) do
    if (C_k = a ∈ A) then
      h_k := {a}      // C_k is an input attribute
    else
      U_k := U;   A_k := ⋃_{C∈prev(C_k)} h_C;
      Apply the RS algorithm to decision table S_{C_k} = (U_k, A_k, dec_{C_k}) to generate
      the rough approximation μ_{C_k} of the concept C_k and μ_{C̄_k} of its complement;
      set h_k(x) := μ_{C_k}, μ_{C̄_k} for all objects x ∈ U;
    end if
  end for
end for
```

4 Experimental Results

We have implemented the proposed solution on the basis of RSES system [2]. To verify a quality of hierarchical classifiers we performed the following experiments.

4.1 Nursery Data Set

This is a real-world model developed to rank applications for nursery schools [7]. The concept ontology is presented in Figure 2. The data set consists of

```
NURSERY                 : not_recom, recommend, very_recom, priority, spec_prior
|⟶ EMPLOY               : Undefined (employment of parents and child's nursery)
|     |⟶ parents        : usual, pretentious, great_pret
|     |⟶ has_nurs       : proper, less_proper, improper, critical, very_crit
| STRUCT_FINAN          : Undefined (family structure and financial standings)
|     |⟶ STRUCTURE:     Undefined (family structure)
|     |     |⟶ form     : complete, completed, incomplete, foster
|     |     |⟶ children : 1, 2, 3, more
|     |⟶ housing        : convenient, less_conv, critical
|     |⟶ finance        : convenient, inconv
|⟶ SOC_HEALTH           : Undefined (social and health picture of the family)
      |⟶ social         : non-prob, slightly_prob, problematic
      |⟶ health         : recommended, priority, not_recom
```

Fig. 2. The ontology of concepts in NURSERY data set

Table 1. Comparison results for Nursery data set achieved on 50% cases for training

	rule-based classifier using original attributes only	Layered learning using intermediate concepts
Classification Accuracy	83.4	99.9%
Coverage	85.3%	100%
Nr of rules	634	42 (for the target concept) 92 (for intermediate concepts)

12960 objects and 8 input attributes which are printed in lowercase. Besides the target concept (NURSERY) the model includes four *undefine intermediate concepts*: EMPLOY, STRUCT_FINAN, STRUCTURE, SOC_HEALTH. To approximate intermediate concepts we have applied a supervised clustering algorithm, in which the similarity between two vectors is determined by a distance between their class distributions. Next, we use rule based algorithm to approximate the target concept. The comparison results are presented in Table 1.

4.2 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 of hierarchical concept approximation problem. Some exemplary concepts and a dependency diagram between those concepts are shown in Fig. 4. Definitions of concepts are given in a form of a question which one can answer YES, NO or NULL (does not concern). We demonstrate the proposed layered learning approach on the simulation system called *road simulator*. The detail description of road simulator has been presented in [6].

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

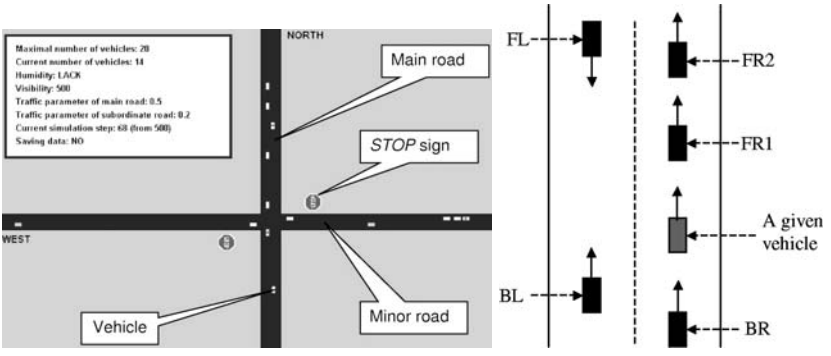


Fig. 3. Left: the board of simulation; Right: given vehicle and five vehicles around him

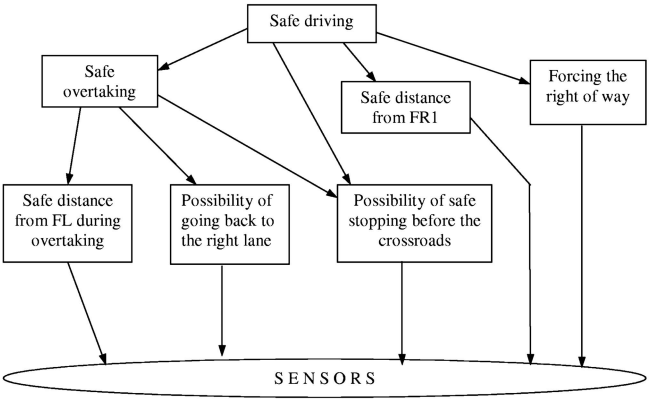


Fig. 4. The relationship diagram for presented concepts

from different devices (sensors) monitoring the situation on the road. During the simulation data may be generated and stored in a text file in a form of a rectangular table (information system). Each line of the table depicts the situation of a single vehicle and contains the sensors' and concepts' values for the vehicle and its neighboring vehicles, see Fig. 3.

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. The comparison results are performed with respect to the accuracy of classification, covering rate of new cases, and computing time necessary for classifier synthesis.

Table 2. Classification accuracy of a standard and hierarchical classifiers

	Accuracy				Coverage			
	Total		Class NO		Total		Class NO	
	RS	**RS-L**	**RS**	**RS-L**	**RS**	**RS-L**	**RS**	**RS-L**
$c10\_s100N$	0.94	0.97	0	0	0.44	0.72	0.50	0.38
$c10\_s200N$	0.99	0.96	0.75	0.60	0.72	0.73	0.50	0.63
$c10\_s300N$	0.99	0.98	0	0.78	0.47	0.68	0.10	0.44
$c10\_s400N$	0.96	0.77	0.57	0.64	0.74	0.90	0.23	0.35
$c10\_s500N$	0.96	0.89	0.30	0.80	0.72	0.86	0.40	0.69
$c20\_s500N$	0.99	0.89	0.44	0.93	0.62	0.89	0.17	0.86
Average	0.97	0.91	**0.34**	**0.63**	**0.62**	**0.79**	**0.32**	**0.55**

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

Tables	**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	98s	104
Average			**90**

Classification Accuracy: Similarly to real life situations, we are interested on the accuracy and the coverage of classifiers on the decision class "safe driving = NO", i.e., dangerous situations. This is a issue for this problem, since datasets are is unbalanced (the concept states only 4% - 9% of training sets).

Table 2 presents the classification accuracy of RS and RS-L classifiers. One can see that 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 training sets reach a sufficient size. The generality of classifiers usually is evaluated by the recognition ability for unseen objects. One can observe the similar scenarios to the accuracy degree. The recognition rate of dangerous situations is very poor in the case of the standard classifier. One can see in Table 2 the improvement on coverage of the hierarchical classifier.

Computing Speed: The layered learning approach also shows a tremendous advantage with respect to the computation time. One can see in Table 3 that speed up ratio of the layered learning approach to the standard one reaches from 40 to 130 times.

5 Conclusion

We presented a new method for improving rough classifiers using concept ontology. 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, which is necessary to created an concept ontology. 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. The advantages of this new approach in comparison to the standard approach have been illustrated by experiments with the road traffic simulator.

Acknowledgements. The research has been partially supported by the grant 3T11C00226 from Ministry of Scientific Research and Information Technology of the Republic of Poland. The authors are deeply grateful to Dr. J. Bazan for his road simulator system and Prof. A. Skowron for valuable discussions on the layered learning approach.

References

1. J. Bazan, H. S. Nguyen, A. Skowron, and M. Szczuka. A view on rough set concept approximation. In Wang G., Liu Q., Yao Y., and Skowron A. (eds), RSFDGrC'2003, Chongqing, China, volume 2639 of *LNAI*, Springer-Verlag Heidelberg 2003, pp. 181–188
2. J. G. Bazan and M. Szczuka. RSES and RSESlib - a collection of tools for rough set computations. Ziarko W. and Yao Y., editors, RSCTC'02, volume 2005 of *LNAI*, pages 106–113, Banff, Canada, October 16-19 2000. Springer-Verlag.
3. Grzymała-Busse J., A New Version of the Rule Induction System LERS Fundamenta Informaticae, Vol. 31(1), 1997, pp. 27–39
4. W. Kloesgen and J. Żytkow, editors. *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, 2002.
5. J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron, Rough sets: A tutorial. In: Pal S.K., Skowron A. (eds.), Rough - fuzzy hybridization: A new trend in decision making, Springer-Verlag, Singapore, 1999, pp. 3–98.
6. S.H. Nguyen, J. Bazan, A. Skowron, and H.S. Nguyen. Layered learning for concept synthesis. In Peters J., Skowron A., Grzymala-Busse J., Kostek B., Swiniarski R., Szczuka M. (eds), *Transactions on Rough Sets I*, volume LNCS 3100 of *Lecture Notes on Computer Science*, Springer, 2004, pp. 187–208.
7. M. Olave, V. Rajkovic and M. Bohanec. An application for admission in public school systems. In I. T. M. Snellen, W. B. H. J. van de Donk and J.-P. Baquiast (eds), Expert Systems in Public Administration, Elsevier Science Publishers (North Holland), 1989, pp. 145–160.
8. S. K. Pal, L. Polkowski, and A. Skowron, editors. *Rough-Neural Computing: Techniques for Computing with Words*. Cognitive Technologies. Springer-Verlag, Heidelberg, Germany, 2003.
9. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*, volume 9 of *System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.

10. Z. Pawlak and A. Skowron. A rough set approach for decision rules generation. In *Proc. of IJCAI'93*, pages 114–119, Chambéry, France, 1993. Morgan Kaufmann.
11. L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *International Journal of Approximate Reasoning*, 15(4):333–365, 1996.
12. L. Polkowski and A. Skowron, editors. *Rough Sets in Knowledge Discovery 1: Methodology and Applications*. Physica-Verlag, Heidelberg, Germany, 1998.
13. A. Skowron. Approximation spaces in rough neurocomputing. In M. Inuiguchi, S. Tsumoto, and S. Hirano, editors, *Rough Set Theory and Granular Computing*, pages 13–22. Springer-Verlag, Heidelberg, Germany, 2003.
14. A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In Pal et al. [8], pages 43–84.
15. P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. The MIT Press, Cambridge, MA, 2000.
16. Wróblewski J., Covering with reducts - a fast algorithm for rule generation. In: Polkowski L., Skowron A.(eds.): Proc. of RSCTC'98, Warsaw, Poland. Springer-Verlag, Berlin, 1998, pp. 402–407.