Rough Set Approach to Approximation of Concepts from Taxonomy

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 hierarchical learning approach to approximation of complex concept from experimental data using concept taxonomy as a given domain knowledge. The proposition is based on rough set and rough mereology theory. 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

Many problems in machine learning and data mining, such as 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, sample) of the universe and using this information one should induce approximations of the concept over the whole universe.

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. 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 taxonomy of concepts. The taxonomy of concepts 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] is an alternative approach to concept approximation.

Given the taxonomy structure, the main idea is to gradually synthesize a target concept from simpler ones. Firstly, the concepts which are located closest to attributes are approximated using feature values available from a data set.

Next, more complex concepts are synthesized from already approximated concepts. This process is repeated for successive layers until the target concept is reached and approximated.

The importance of hierarchical concept synthesis is now well recognized by researchers (see, e.g., [10, 7]). An idea of hierarchical concept synthesis, in the rough mereological and granular computing frameworks has been developed (see, e.g., [10, 7, 11]) and problems connected with compound concept approximation are discussed, e.g., in [7, 12, 1, 14].

In this paper we concentrate on concepts that are specified by decision classes in decision systems [8]. 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. In this paper we discuss some strategies for concept composing founded on the rough set theory approach. We also examine effectiveness of layered learning approach by comparison with standard rule-based learning approach. Quality of the new approach will be verified with respect to generality of concept approximation, preciseness of concept approximation, computation time required for concept induction and concept description lengths. Experiments are carried out on well-known benchmark data set as well as on an artificial data set generated by a road traffic simulator.

2 Basic notions

The problem of concept approximation can be formulated in many ways. In this paper we treat it as a problem of searching for description of an unknown concept. The description must be expressible in a known language.

Formally, given an universe $\mathcal{X}$ of objects (cases, states, patients, observations, etc.), 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 $a : \mathcal{X} \to V_a$ 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 $x \in \mathcal{X}$, 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 concepts, 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, ..., which can be also estimated by so called testing 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., $U = CLASS_1 \cup \ldots \cup CLASS_d$. In case of concept approximation problem, we have $U = CLASS_{yes} \cup CLASS_{no}$.

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

3.1 Preliminaries

Rough set methodology for concept approximation can be described as follows. Let $C \subseteq \mathcal{X}$ be a concept and let $U \subseteq \mathcal{X}$ be a finite sample of $\mathcal{X}$. Assume that for any $x \in U$ there is given information if $x \in C \cap U$ or $x \in U - C$. Any pair $\mathbb{P} = (\mathbf{L}, \mathbf{U})$ is called *rough approximation of* C (see [1, 8]) 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) 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 C, respectively. The set $\mathbf{BN} = \mathbf{U} - \mathbf{L}$ is called the *boundary region of approximation* of C. For objects $x \in \mathbf{L}$, we say that "surely, x is in 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.

The rough approximation of concept can be also defined by means of 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.

Many methods of discovering rough approximations of concepts from data have been proposed, e.g., method based on reducts [8][9], on k-NN classifiers [1], or on decision rules [1]. Let us remind the construction of rough membership function in the concept approximation approach based on decision rules.

Let $\mathbb{S} = (U, A, dec)$ be a given decision table. Any implication of the form:

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

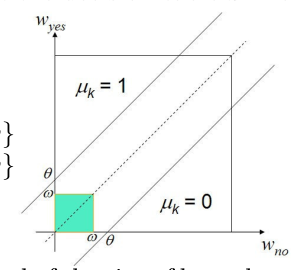
where $a_{i_j} \in A$ and $v_j \in V_{a_{i_j}}$, is called a *decision rule* for the k^{th} decision class. Let $\mathbf{RULES}(\mathbb{S})$ is a set of decision rules induced from $\mathbb{S}$ by some rule extraction method. For any object $x \in \mathcal{X}$, let $MatchRules(\mathbb{S}, x)$ be the set of rules from $\mathbf{RULES}(\mathbb{S})$ supported by x. One can define the rough membership function $\mu_k : \mathcal{X} \to [0, 1]$ 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 $\mathbf{R}_{no}$ 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*, $confidence$ of $\mathbf{r}$ and some global information about the decision table $\mathbb{S}$ like table size, class distribution (see [2]).
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 Layered learning method based on rough sets

In this section we describe a strategy that learns to approximate the concept established on the higher level of a given taxonomy 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 taxonomy 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;
– $\mathcal{R} \subset \mathcal{C} \times \mathcal{C}$ 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 (based on rough set theory) to concept approximation problem. Let us denote this method by RS algorithm. This algorithm has been designed for *flat* hierarchy of concepts (i.e., the target concept (decision attribute) is connected directly to input attributes). Let us remind the specification of RS algorithm:

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 attributes h_C that indicates the membership of any object x (x not necessary belongs to U) to the concept C. There are two possible models
1. Voting weights: $h_C(x) = \{w^C_{yes}(x), w^C_{no}(x)\}$
2. Rough membership functions: $h_C(x) = \{\mu_C(x)\}$

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 $prev(C) = \{C_1, C_2, ..., C_m\}$ be the set of concepts in the lower layer, which are connected with concept 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 $\mathbb{S}_C$;

Hence the main trouble is based on construction of the 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, then $h_{C_i}(x) = \{C_i(x)\}$, otherwise $h_{C_i}(x) = app_{C_i}(x)$ (see Fig. 1).

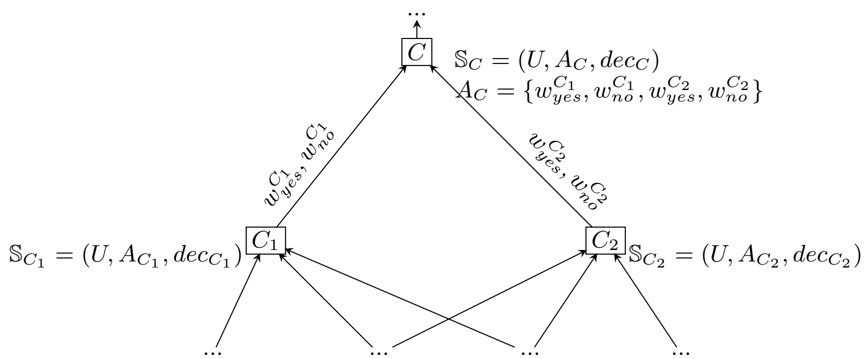


Fig. 1. The construction of decision table for a higher level concept using rough approximation of concepts from lower level

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 taxonomy. 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 ($l == 0$) **then**
      $h_k := \{C_k\}$ // *C_k is an input attribute*
    else
      $U_k := U$;
      $A_k = \bigcup_{C \in prev(C_k)} h_C$;
      Apply the RS algorithm to decision table $\mathbb{S}_{C_k} = (U_k, A_k, dec_{C_k})$ (described in Section 3.1) to generate the rough approximation of app_{C_k} of concept C_k
      set $h_k(x) := app_{C_k}(x)$ for all objects $x \in U$;
    end if
  end for
end for

The most advanced feature of the proposed method is the possibility of tuning the quality of concept approximation process via the parameters $omega_C, \theta_C$. More details about this problem will be discussed in our next contribution.

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)

4 Experimental Results

We have implemented the proposed solution on the basis of RSES system [3]. 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 [6]. The taxonomy of concepts is presented in Figure 2. The data set consists of 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. This data set is interesting for our consideration, because the values of intermediate concepts are unknown. We applied a clustering algorithm to approximate intermediate concepts. Next, we use rule based algorithm (in RSES system) to approximate the target concept. The comparison results are presented in Table 1.

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 taxonomy of concepts in NURSERY data set

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. 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 most important features of this system.

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 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 based on the sensor's values in a way determined by an expert. These parameters in the present simulator version take over the binary values and are therefore called concepts. Concepts definitions are very often in a form of a question which one can answer YES, NO or NULL (does not concern). In Figure 4 there is an exemplary relationship diagram for the above mentioned concepts we present some exemplary concepts and dependency diagram between those concepts.

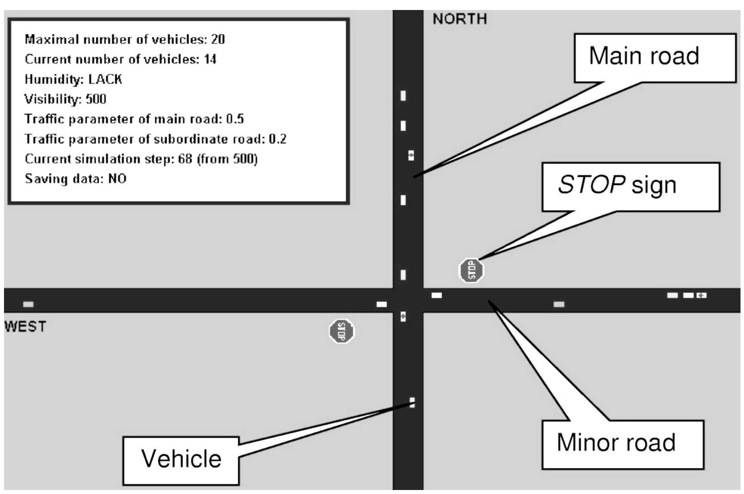


Fig. 3. the board of simulation

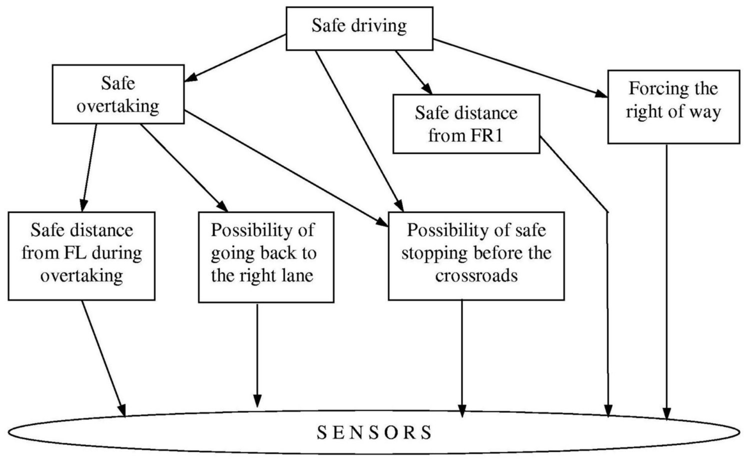


Fig. 4. The relationship diagram for presented concepts

During the simulation data may be generated and stored in a text file in a form of a rectangle board (information system). Each line of the board depicts the situation of a single vehicle and the sensors' and concepts' values are registered for a given vehicle and its neighboring vehicles. Within each simulation step descriptions of situations of all the vehicles are saved to file.

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 [3] 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 4): 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.*"

These tables are created using information available from the concept relationship diagram presented in Figure 4. A concept in the next level is C_6: "*safe overtaking*". To approximate concept C_6, we create a table with three conditional attributes. These attributes describe fitting degrees of object to concepts C_1, C_2, C_3, respectively. The decision attribute has three values YES, NO, or

$NULL$ corresponding to the cases of safe overtaking, dangerous overtaking, and not applicable (the overtaking has not been made by car). 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. The decision attribute has two possible values YES or NO if a car is satisfying global safety condition, or not, respectively.

The comparison results are performed with respect to the following criteria: (1) Accuracy of classification, (2) Covering rate of new cases (generality), and (3) Computing time necessary for classifier synthesis.

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. In experiments we concentrate on quality of the "NO" class approximation.

In Table 2 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 training sets reach a sufficient size.

Table 2. Classification accuracy of a 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**

Covering rate: Generality of classifiers usually is evaluated by the recognition ability for unseen objects. In this section we analyze covering rate of classifiers for new objects. One can observe the similar scenarios to the accuracy degree. The recognition rate of situations belonging to "NO" class is very poor in the case of the standard classifier. One can see in Table 3 the improvement on coverage degree of "YES" class and "NO" class of the hierarchical classifier.

Computing speed: With time respect the layered learning approach shows a tremendous advantage in comparison with the standard learning approach. In

Table 3. Covering rate for 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**

the case of standard classifier, computational time is measured as a time required for computing a rule set using to decision class approximation. In the case of hierarchical classifier computational time is a total time required for all sub-concepts and target concept approximation. One can see in Table 4 that speed up ratio of the layered learning approach to the standard one reaches from 40 to 130 times (all experiments were performed on computer with processor AMD Athlon 1.4GHz., 256MB RAM)

Table 4. Time for standard and hierarchical classifier generation

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**

5 Conclusion

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.

Acknowledgements: The research has been supported by the 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 G. Wang, Q. Liu, Y. Yao, and A. Skowron, editors, RSFD-GrC'2003, Chongqing, China, volume 2639 of *LNAI*, pages 181–188, Heidelberg, Germany, 2003. Springer-Verlag.
2. J. G. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In L. Polkowski and A. Skowron, editors, *Rough Sets in Knowledge Discovery 1: Methodology and Applications*, pages 321–365. Physica-Verlag, Heidelberg, Germany, 1998.
3. J. G. Bazan and M. Szczuka. RSES and RSESlib - a collection of tools for rough set computations. In W. Ziarko and Y. Yao, editors, RSCTC'02, volume 2005 of *LNAI*, pages 106–113, Banff, Canada, October 16-19 2000. Springer-Verlag.
4. W. Kloesgen and J. Żytkow, editors. *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, 2002.
5. T. Mitchell. *Machine Learning*. Mc Graw Hill, 1998.
6. 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.
7. S. K. Pal, L. Polkowski, and A. Skowron, editors. *Rough-Neural Computing: Techniques for Computing with Words*. Cognitive Technologies. Springer-Verlag, Heidelberg, Germany, 2003.
8. 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.
9. 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.
10. L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *International Journal of Approximate Reasoning*, 15(4):333–365, 1996.
11. A. Skowron. Approximate reasoning by agents in distributed environments. In N. Zhong, J. Liu, S. Ohsuga, and J. Bradshaw, editors, IAT'01, Japan, pages 28–39. World Scientific, Singapore, 2001.
12. 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.
13. A. Skowron and J. Stepaniuk. Information granule decomposition. *Fundamenta Informaticae*, 47(3-4):337–350, 2001.
14. A. Skowron and M. Szczuka. Approximate reasoning schemes: Classifiers for computing with words. In *Proceedings of SMPS 2002*, Advances in Soft Computing, pages 338–345, Heidelberg, Canada, 2002. Springer-Verlag.
15. P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. The MIT Press, Cambridge, MA, 2000.