Ontology Driven Concept Approximation

Sinh Hoa Nguyen[1], Trung Thanh Nguyen[2], and Hung Son Nguyen[3]

[1] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008, Warsaw, Poland
[2] Department of Computer Science, University of Bath
Bath BA2 7AY, United Kingdom
[3] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
son@mimuw.edu.pl

Abstract. This paper investigates the concept approximation problem using ontology as an domain knowledge representation model and rough set theory. In [7] [8], we have presented a rough set based multi-layered learning framework for approximation of complex concepts assuming the existence of a simple concept hierarchy. The proposed methodology utilizes the ontology structure to learn compound concepts using the rough approximations of the primitive concepts as input attributes. In this paper we consider the extended model for knowledge representation where the concept hierarchies are embedded with additional knowledge in a form of relations or constrains among sub-concepts. We present an extended multi-layered learning scheme that can incorporate the additional knowledge and propose some classes of such relations that assure an improvement of the learning algorithm as well as a convenience of the knowledge modeling process. We illustrate the proposed method and present some results of experiment with data from sunspot recognition problem.

Keywords: ontology, concept hierarchy, rough sets, classification.

1 Introduction

In AI, approximate reasoning is a crucial problem occurring, e.g., during an interaction between two intelligent (human/machine) beings which are using different languages to talk about objects from the same universe. The intelligence skill of those beings (also called agents) is measured by the ability of understanding the other agents. This skill appears in different ways, e.g., as a learning or classification in machine learning and pattern recognition theory, or as an adaptation in evolutionary computation theory. A great effort of researchers in machine learning and data mining has been made to develop efficient methods for approximation of concepts from data [6]. Nevertheless, there exist many problems that are still unsolvable for existing state of the art methods, because of the high complexity of learning algorithms or even unlearnability of hypothesis spaces.

Utilization of domain knowledge into learning process becomes a big challenge for improving and developing more efficient concept approximation methods. In previous papers we have assumed that the domain knowledge was given in form

S. Greco et al. (Eds.): RSCTC 2006, LNAI 4259, pp. 547–556, 2006.

of a concept hierarchy [7] [8]. The concept hierarchy, the simplest form of ontology, is a treelike structure with the target concept located at the root, with attributes located at leaves, and with some additional concepts located in internal nodes. We have adopt the layered learning approach [13], and rough set methods to proposed a multi-layered algorithm for induction of "multi-layer rough classifier" (MLRC) from data [7]. We have shown that MLRC has significantly better classification accuracy and shorter classification time comparing with the traditional rough classifiers. Nevertheless, many problems still remain in this research. The problem is related to the choice of the appropriate learning algorithm and the corresponding decision table for approximation of each concept in the hierarchy. Moreover, during experiment execution, we observed a noticeable worsening of accuracy of classifiers in the consecutive layers. This is because, except the own approximation error, the compound classifier can have a mistake even when only one of its component classifiers fails, e.g., it has a misclassification or returns no answer.

The mentioned above problems are probably caused by the simplification of the knowledge representation model, where the only structure of concept ontology was utilized in the learning algorithm. In this paper we consider an extended knowledge representation model, where except the concept hierarchy, we assume that there are some constraints between concepts on the same level. We will present a modified layered learning algorithm that utilizes those constraints as the additional domain knowledge.

The paper is organized as follows. Section 2.2 provides some basic notions of concept ontology, some important classes of concept relations and some basic ideas of rough set theory and the problem of concept approximation. Section 3 presents a multi-layered learning algorithm driven by ontology for the concept approximation problem. Section 4 is devoted to illustration and analyzing the accuracy of the proposed method for the sunspot recognition problem. The paper finishes with summarized conclusions and discussion on possible feature works.

2 Preliminaries

Concepts can be understood as definable sets of objects. Formally, any subset X of a given universe $\mathfrak{U}$ which can be described by a formula of $\mathcal{L}$ is called the concept in $\mathcal{L}$. The *concept approximation problem* can be understood as searching for approximate description – using formulas of a predefined language $\mathcal{L}$ – of concepts that are definable in other language $\mathcal{L}^*$. Inductive learning is the concept approximation method that searches for description of unknown concept using finite set $U \subset \mathfrak{U}$ of training examples.

2.1 The Role of Ontology in Inductive Learning

Ontology is defined in literature as a formal description of concept names and relation types organized in a partial ordering by the concept-subconcept relation [12]. Syntactically, given a logical language $\mathcal{L}$, an ontology is a tuple $\langle V, A \rangle$, where the vocabulary V is a subset of the predicate symbols of $\mathcal{L}$ and the axioms A are a subset of the well-formed formulas of $\mathcal{L}$ [5]. The set V is interpreted as a set of concepts and the set A is a set of relations among concepts present in the set V. A taxonomy is the most commonly used form of ontologies. It is usually a hierarchical classification of concepts in the domain, therefore we would draw it in the form of tree and call it a concept hierarchy.

Nowadays, ontology is used as an alternative knowledge representation model, and it becomes a hot topic in many research areas including (1) ontological specification for software development (2) ontology driven information systems (3) ontology-based semantic search (4) ontology-based knowledge discovery and acquisition [3] [4]. Many applications in data mining make use of taxonomies to describe different levels of generalization of concepts defined within attribute domains [5]. The role of taxonomies is to guide a pattern extraction process to induce patterns in different levels of abstractions.

Ontologies are also useful for concept approximation problems in another context. One can utilized the concept hierarchy describing the relationship between the target concept (defined by decision attribute) and conditional attributes (through additional concepts if necessary) in the induction process. Such hierarchy can be exploited as a guide to decomposition of complex concept approximation problem into simpler ones and to construction of compound classifiers for the target concept from the classifiers for primitive concepts [15].

2.2 Rough Sets and Rough Classifiers

Rough set theory has been introduced by Professor Z. Pawlak [9] as a tool for approximation of concepts under uncertainty. The theory is featured by operating on two definable subsets, i.e., a lower approximation and upper approximation. The first definition, so called the "standard rough sets", was introduced by Pawlak in his pioneering book on rough set theory [9].

Given an information system $\mathbb{S} = (U, A)$, where U is the set of training objects, A is the set of attributes and a concept $X \subset U$. Assuming at the moment that only some attributes from $B \subset A$ are accessible, then this problem can be also described by appropriate decision table $\mathbb{S} = (U, B \cup \{dec_X\})$, where $dec_X(u) = 1$ for $u \in X$, and $dec_X(u) = 0$ for $u \notin X$.

First one can define called the *B-indiscernibility relation* $IND(B) \subset U \times U$ in such a way that $x\ IND(B)\ y$ if and only if x, y are indiscernible by attributes from B, i.e., $inf_B(x) = inf_B(y)$. Let $[x]_{IND(B)} = \{u \in U : (x, u) \in IND\,(B)\}$ denote the equivalence class of $IND\,(B)$ defined by x. The lower and upper approximations of X (using attributes from B) are defined by:

$$\mathbf{L}_B(X) = \left\{x : [x]_{IND(B)} \subseteq X\right\}; \quad \mathbf{U}_B(X) = \left\{x : [x]_{IND(B)} \cap X \neq \varnothing\right\}$$

Let us point out that there are many extensions of the standard definition of rough sets, e.g., variable rough set model [14] or tolerance approximation space [11]. In these methods, rough approximations of concepts can be also defined by *rough membership function*, i.e., a mapping $\mu_X : U \to [0, 1]$ such that $\mathbf{L}_{\mu_X} = \{x \in U : \mu_C(x) = 1\}$ and $\mathbf{U}_{\mu_X} = \{x \in U : \mu_X(x) > 0\}$ are lower and upper approximation of a given concept X. In case of the classical rough set theory, the rough membership function is defined by $\mu_X^B(x) = \frac{|X \cap [x]_{IND(B)}|}{|[x]_{IND(B)}|}$.

The inductive learning approach to rough approximations of concepts we assume that U is a finite sample of objects from a universe $\mathfrak{U}$ and $X = \mathcal{C} \cap U$ is the representation of a unknown concept $\mathcal{C} \subset \mathfrak{U}$ in U. The problem can be understood as searching for an extended rough membership function $\mu_{\mathcal{C}} : \mathfrak{U} \to [0,1]$ for $\mathcal{C} \subset \mathfrak{U}$ such that the corresponding rough approximations defined by $\mu_{\mathcal{C}}$ are the good approximations of $\mathcal{C}$.

$$\begin{array}{lcl} U & \dashrightarrow & \mu_X : U \to [0,1] \\ \cap & & \Downarrow \\ \mathfrak{U} & \dashrightarrow & \mu_{\mathcal{C}} : \mathfrak{U} \to [0,1] \end{array}$$

The algorithm that calculates the value $\mu_{\mathcal{C}}(x)$ of extended rough membership function for each new unseen object $x \in \mathfrak{U}$ is called *the rough classifier*. In fact, rough classifiers can be constructed by *fuzzification* of other classifiers [1]. The specification of each algorithm for induction of rough classifiers is as follows:

Input: Given a decision table $\mathbb{S}_C = (U, A_C, dec_C)$
Output: Approximation of C in form of a hypothetical classifier $h_C = \{\mu_C, \mu_{\overline{C}}\}$ indicating the membership $\mu_C(x)$ of any object $x \in \mathfrak{U}$ to the concept C or the membership $\mu_{\overline{C}}(x)$ to its complement $\overline{C}$.

Rule-based, kNN-based and decision tree based rough classifiers are examples of rough classifier types that will be used in next sections. These methods will be used as building blocks for construction of compound classifiers.

3 Ontology Driven Construction of Rough Classifiers

Induction of rough classifiers is the most important step in many applications of rough set theory in the process of knowledge discovery from databases. We have presented a multi-layered learning scheme for approximation of complex concept assuming that a hierarchy of concepts is given. The main idea is to gradually synthesize the target concept from the simpler ones. At the lowest layer, basic concepts are approximated using input features available from the data set. At the next layers, the approximations of compound concepts are synthesized using rough approximations of concepts from the previous layer. This process is repeated for successive layers and it results in the creation of a multi-layer rough classifier (MLRC). The advantages of MLRC have been recognized and confirmed by many experiments over different concept approximation problems [7] [8]. But in case of poor quality (incomplete, noisy) data sets, this learning scheme gives approximations with unsatisfactory accuracy because of the high sensitiveness of compound rough classifiers.

In this paper, except the concept hierarchy, we propose to extend the knowledge representation model by some constraints between concepts on the same level. We show that such constraints can improve the quality of classifiers.

3.1 Knowledge Representation Model with Constraints

Recall that taxonomy, or concept hierarchy, represents a set of concepts and a binary relation which connects a "child" concept with its "parent". The most important relation types are the subsumption relations (written as "is-a" or "is-part-of") defining which objects (or concepts) are members (or parts) of another concepts in the ontology. This model facilitates the user to represent his/her knowledge about relationships between input attributes and target concepts. If no such information available, one can assume the flat hierarchy with the target concept on top and all attributes in the downstairs layer. Besides the "child-parent" relations, we proposed to associate with each parent concept a set of "domain-specific" constraints. We consider two types of constraints: (1) constraints describing relationships between a concept and its sub-concepts; and (2) constraints connecting the "sibling" concepts (having the same parent).

Formally, the extended concept hierarchy is a triple $\mathcal{H} = (\mathbb{C}, \mathbb{R}, Constr)$, where $\mathbb{C} = \{C_1, ..., C_n\}$ is a finite set of concepts including basic concepts (attributes), intermediated concepts and target concept; $\mathbb{R} \subseteq \mathbb{C} \times \mathbb{C}$ is child-parent relation in the hierarchy; and $Constr$ is a set of constraints. In this paper, we consider constraints expressed by association rules of the form $\mathbf{P} \to_\alpha \mathbf{Q}$, where

- $\mathbf{P}, \mathbf{Q}$ are boolean formulas over the set $\{c_1, ..., c_n, \overline{c_1}, ..., \overline{c_n}\}$ of boolean variables corresponding to concepts from $\mathbb{C}$ and their complements;
- $\alpha \in [0, 1]$ is the confidence of this rule;

In next sections, we will consider only two types of constraints, i.e., the *"children-parent"* type of constraints connecting some "child" concepts with their common parent, and the *"siblings-sibling"* type of constraints connecting some sibling concepts with another sibling.

3.2 Learning Algorithm for Concept Approximation

Let us assume that an extended concept hierarchy $\mathcal{H} = (\mathbb{C}, \mathbb{R}, Constr)$ is given. For compound concepts in the hierarchy, we can use the rough classifiers as a building blocks to develop a multi-layered classifier. More precisely, let $prev(C) = \{C_1,..., C_m\}$ be the set of concepts, which are connected with C in the hierarchy. The rough approximation of the concept C can be determined by performing two steps: (1) construct a decision table $\mathbb{S}_C = (U, A_C, dec_C)$ relevant for the concept C; and (2) induce a rough classifier for C using decision table $\mathbb{S}_C$. In [7], the training table $\mathbb{S}_C = (U, A_C, dec_C)$ is constructed as follows:

- The set of objects U is common for all concepts in the hierarchy.
- $A_C = h_{C_1} \cup h_{C_2} \cup ... \cup h_{C_m}$, where h_{C_i} is the output of the hypothetical classifier for the concept $C_i \in prev(C)$. If C_i is an input attribute $a \in A$ then $h_{C_i}(x) = \{a(x)\}$, otherwise $h_{C_i}(x) = \{\mu_{C_i}(x), \mu_{\overline{C_i}}(x)\}$.

Repeating those steps for each concept through the bottom to the top layer we obtain a "hybrid classifier" for the target concept, which is a combination of classifiers of various types. In the second step, the learning algorithm should use the decision table $\mathbb{S}_C = (U, A_C, dec_C)$ to "resolve conflicts" between classifiers of its children. One can observe that, if sibling concepts $C_1, ..., C_m$ are independent, the membership function values of these concepts are "sent" to the "parent" C, without any correction. Thus the membership value of weak classifiers may disturb the training table for the parent concept and cause the misclassification when testing new unseen objects. We present two techniques that enable the expert to improve the quality of hybrid classifiers by embeding their domain knowledge into learning process.

1. Using Constraints to Refine Weak Classifiers: Let $R := c_1 \wedge c_2 ... \wedge c_k \rightarrow_\alpha c_0$ be a siblings-sibling constraint connecting concepts $C_1, ..., C_k$ with the concept C_0.We say that the constraint R *fires* for C_0 if

- Classifiers for concepts $C_1, ..., C_k$ are strong (of a high quality).
- Classifier of concepts C_0 is weak (of a low quality).

The refining algorithm always starts with the weakest classifier for which there exist a constraint that fires (see Algorithm 1).

Algorithm 1. Classifier Refining

Input: classifier $h(C_0)$, constraint $R := c_1 \wedge c_2 ... \wedge c_k \rightarrow_\alpha c_0$
Output: Refined classifier $h(C_0)$

```
for each object x ∈ U do
    if x are recognized by classifiers of C_1, ..., C_k with high degree then
        if c_0 is a positive literal then
            μ_{C_0}(x) := α · min{μ_{C_1}(x), μ_{C_2}(x), ..., μ_{C_k}(x)};   μ_{\overline{C_0}}(x) := 1 − μ_{C_0}(x);
        else {c_0 is a negative literal}
            μ_{\overline{C_0}}(x) := α · min{μ_{C_1}(x), μ_{C_2}(x), ..., μ_{C_k}(x)};   μ_{C_0}(x) := 1 − μ_{\overline{C_i}}(x)
    h'(C_j) := (μ_{C_0}, μ_{\overline{C_0}});
```

2. Using Constraints to Select Learning Algorithm: Another problem is how to assign a suitable approximation algorithm for an individual concept in the concept hierarchy? In the previous papers [7] the type of approximation algorithm (knn, decision tree or rule set) for each concept was settled by the user. In this paper we show that the constraints can be treated as a guide to semi-automatic selection of best learning algorithms for concepts in the hierarchy.

Assume that there is a "children-parent" constraints: $\bigwedge_i c_i \rightarrow_\alpha p$ (or $\bigwedge_i c_i \rightarrow_\alpha \overline{p}$) for a concept $P \in \mathbb{C}$. The idea is to choose the learning algorithm that maximizes the confidence of constraints connecting $P's$ children with himself. Let **RS_ALG** be a set of available parameterized learning algorithms, we define an objective function $\Psi_P : \mathbf{RS\_ALG} \rightarrow \Re^+$ to evaluate the quality of algorithms. For each algorithm $\mathbf{L} \in \mathbf{RS\_ALG}$ the value of $\Psi_P(\mathbf{L})$ is depended on two factors:

- Classification quality of $\mathbf{L}(\mathbb{S}_P)$ on a validation set of objects;
- Confidence of the constraints $\bigwedge_i c_i \rightarrow_\alpha p$

The function $\Psi_P(\mathbf{L})$ should be increasing w.r.t. quality of the classifier $\mathbf{L}(\mathbb{S}_P)$ for the concept P (induced by $\mathbf{L}$) and the closeness between the real confidence of the association rule $\bigwedge_i c_i \rightarrow p$ and the parameter α. The function Ψ can be used as an objective function to evaluate a quality of approximation algorithm.

Algorithm 2. Induction of multi-layered rough classifier using constraints

Input: Decision system $\mathbb{S} = (U, A, d)$, extended concept hierarchy $\mathcal{H} = (\mathbb{C}, \mathbb{R}, Constr)$; a set **RS_ALG** of available approximation algorithms
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
            U_k := U;
            A_k := B, where B ⊆ A is a set relevant to define C_k
        else
            U_k := U
            A_k = ⋃ O_i, for all C_i ∈ prev(C_k), where O_i is the output vector of C_i;
            Choose the best learning algorithm L ∈ RS_ALG with a maximal objective
            function Ψ_{C_k}(L)
            Generate a classifier H(C_k) of concept C_k;
            Refine a classifier H(C_k) using a constraint set Constr.
            Send output signals O_k = {μ_C(x), μ_{C̄}(x)} to the parent to the next level.
```

A complete scheme of multi-layered learning algorithm with concept constrains is presented in Algorithm 2.

4 Example and Experimental Results

Sunspots are the subject of interest to many astronomers and solar physicists. Sunspot observation, analysis and classification form an important part of furthering the knowledge about the Sun. Sunspot classification is a manual and very labor intensive process that could be automated if successfully learned by a machine. The main goal of the first attempt to sunspot classification problem is to classify sunspots into one of the seven classes $\{A, B, C, D, E, F, H\}$, which are defined according to the McIntosh/Zurich Sunspot Classification Scheme. More detailed description of this problem can be found in [8].

The data was obtained by processing NASA SOHO/MDI satellite images to extract individual sunspots and their attributes characterizing their visual properties like size, shape, positions. The data set consists of 2589 observations from the period of September 2001 to November 2001. The main difficulty in correctly determining sunspot groups concerns the interpretation of the classification scheme itself. There is a wide allowable margin for each class (see Figure 1). Therefore, classification results may differ between different astronomers doing the classification.

Now, we will present the application of the proposed approach to the problem of sunspot classification. In [8], we have presented a method for automatic

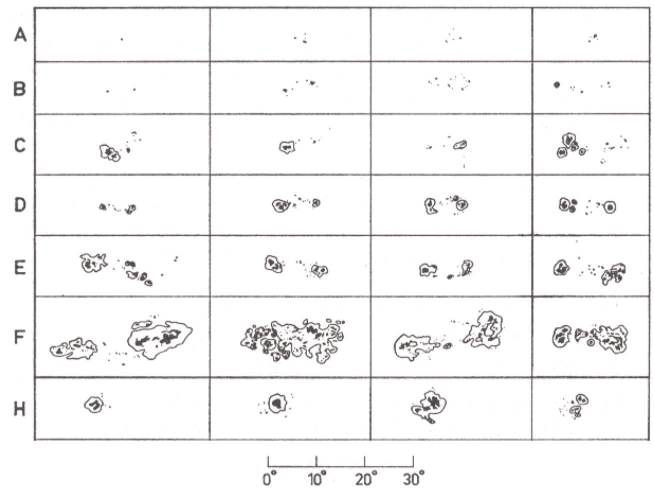

Fig. 1. Possible visual appearances for each class. There is a wide allowable margin in the interpretation of the classification rules making automatic classification difficult.

modeling the domain knowledge about sunspots concept hierarchy. The main part of this ontology is presented in Figure 2.

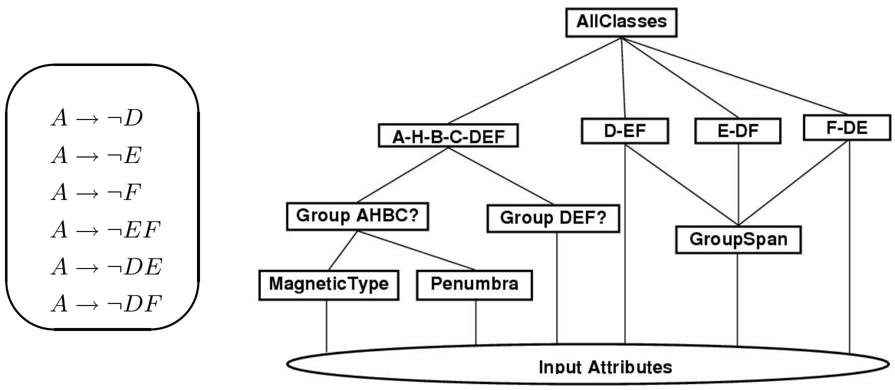

Fig. 2. The concept hierarchy for sunspot recognition problem

We have shown that rough membership function can be induced using different classifiers, e.g., k-NN, decision tree or decision rule set. The problem is to chose the proper type of classifiers for every node of the hierarchy. In experiments with sunspot data, we applied the rule based approach for concepts in the lowest level, decision tree based approach for the concepts in the intermediate levels and the nearest neighbor based approach the target concept.

Figure 3 (left) presents the classification accuracy of "hybrid classifier" obtained by composition of different types of classifiers and "homogenous classifier" obtained by composition of one type of classifiers. The first three bars show qualities of homogenous classifiers obtained by composition of k-NN classifiers, decision tree classifiers and rule based classifiers, respectively. The fourth bar (the gray one) of the histogram displays the accuracy of the hybrid classifier.

The use of constraints also give a profit. In our experiment, these constraints are defined for concepts at the second layer to define the training table for the target concept *AllClasses*. It is because the noticeable breakdown of accuracy have been observed during experiments. We use the strategy proposed in Section 3 to settle the final rough membership values obtained from its children *A-H-B-C-DEF*, *D-EF*, *E-DF*, *F-DE* (see the concept hierarchy). One can observe that using constraints we can promote good classifiers in a composition step. A better classifier has higher priority in a conflict situation. The experiment results are shown in Figure 3. The gray bar of the histogram displays the quality of the classifier induced without concept constraints and the black bar shows the quality of the classifier generated using additional constraints.

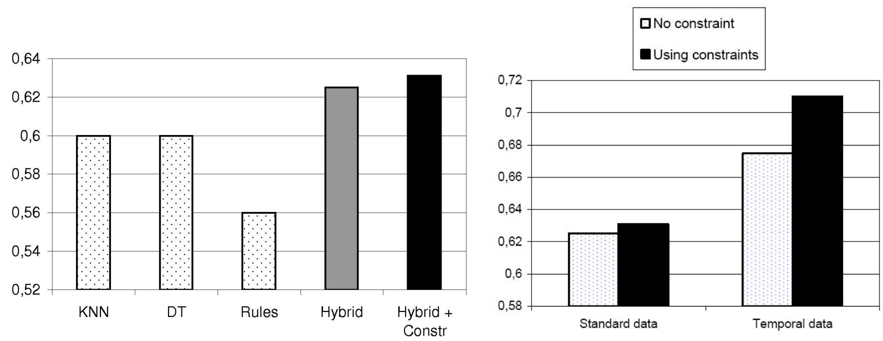

Fig. 3. Accuracy comparison of different layered learning methods

Another approach to manage with sunspot recognition problem is related to temporal features. Comparative results are showed in Figure 3 (right). The first two bars in the graph describe the accuracy of classifiers induced *without* temporal features and the last two bars display the accuracy of classifiers induced *with* temporal features. One can observe a clear advantage of the last classifiers over the first ones. The experimental results also show that the approach for dealing with temporal features and concept constraints considerably improves approximation quality of the complex groups such as B, D, E and F.

5 Conclusions

We presented some extensions of a layered learning approach. Unlike traditional approach, in the layered learning approach the concept approximations are induced not only from available data sets but also from expert's domain knowledge. In the paper, besides a concept dependency hierarchy we have also considered additional domain knowledge in the form of concept constraints. We proposed an approach to deal with some forms of concept constraints. Experimental results with sunspot classification problem have shown advantages of these new approaches in comparison to the standard learning approach.

Acknowledgement. The research has been partially supported by the grant 3T11C00226 from Ministry of Scientific Research and Information Technology of the Republic of Poland and the research grant of Polish-Japanese Institute of Information Technology.

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, RSFDGrC'2003, Chongqing, China, volume 2639 of *LNAI*, pages 181–188, Heidelberg, Germany, 2003. Springer-Verlag.
2. J. Bazan, M. Szczuka. RSES and RSESlib - A Collection of Tools for Rough Set Computations, Proc. of RSCTC'2000, LNAI 2005, Springer Verlag, Berlin, 2001
3. J. Davies, D. Fensel and F. van Harmelen (eds), Towards the Semantic Web – Ontology-Driven Knowledge Management. Wiley, London, UK, 2002.
4. A. Gomez-Perez, O. Corcho, M. Fernandez-Lopez. Ontological Engineering, Springer-Verlag, London, Berlin, 2002.
5. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2000.
6. W. Kloesgen and J. Żytkow, editors. *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, 2002.
7. S.H. Nguyen, J. Bazan, A. Skowron, and H.S. Nguyen. *Layered learning for concept synthesis*. Jim F. Peters, A. Skowron, J.W. Grzymala-Busse, B. Kostek, R.W.Swiniarski, and M. S. Szczuka, editors, Transactions on Rough Sets I, LNCS 3100, pp. 187-208. Springer, 2004.
8. S.H. Nguyen, T.T. Nguyen, and H.S. Nguyen. *Rough Set Approach to Sunspot Classification Problem*. In Proc. of Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2005), Part II, Regina, Canada, August/September 2005, pp. 263-272. Springer, 2005.
9. Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11:341–356, 1982.
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. 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.
12. J. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA (2000)
13. P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. The MIT Press, Cambridge, MA, 2000.
14. W. Ziarko. Variable precision rough set model. *Journal of Computer and System Sciences*, 46:39–59, 1993.
15. B. Zupan, M. Bohanec, I. Bratko, and J. Demsar, "Machine learning by function decomposition," in Proc. Fourteenth International Conference on Machine Learning, (San Mateo, CA), pp. 421–429, Morgan Kaufmann, 1997