Rough Sets and Vague Concept Approximation: From Sample Approximation to Adaptive Learning

Jan Bazan[1], Andrzej Skowron[2], and Roman Swiniarski[3,4]

[1] Institute of Mathematics, University of Rzeszów
Rejtana 16A, 35-959 Rzeszów, Poland
bazan@univ.rzeszow.pl
[2] Institute of Mathematics
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[3] Institute of Computer Science, Polish Academy of Sciences
Ordona 21, 01-237 Warsaw, Poland
and
Department of Mathematical and Computer Sciences
San Diego State University
5500 Campanile Drive San Diego, CA 92182, USA
rswiniar@sciences.sdsu.edu

Abstract. We present a rough set approach to vague concept approximation. Approximation spaces used for concept approximation have been initially defined on samples of objects (decision tables) representing partial information about concepts. Such approximation spaces defined on samples are next inductively extended on the whole object universe. This makes it possible to define the concept approximation on extensions of samples. We discuss the role of inductive extensions of approximation spaces in searching for concept approximation. However, searching for relevant inductive extensions of approximation spaces defined on samples is infeasible for compound concepts. We outline an approach making this searching feasible by using a concept ontology specified by domain knowledge and its approximation. We also extend this approach to a framework for adaptive approximation of vague concepts by agents interacting with environments. This paper realizes a step toward approximate reasoning in multiagent systems (MAS), intelligent systems, and complex dynamic systems (CAS).

Keywords: Vagueness, rough sets, approximation space, higher order vagueness, adaptive learning, incremental learning, reinforcement learning, constraints, intelligent systems.

1 Introduction

In this paper, we discuss the rough set approach to vague concept approximation. There has been a long debate in philosophy about vague concepts [18].

J.F. Peters and A. Skowron (Eds.): Transactions on Rough Sets V, LNCS 4100, pp. 39–62, 2006.

Nowadays, computer scientists are also interested in vague (imprecise) concepts, e.g, many intelligent systems should satisfy some constraints specified by vague concepts. Hence, the problem of vague concept approximation as well as preserving vague dependencies especially in dynamically changing environments is important for such systems. Lotfi Zadeh [66] introduced a very successful approach to vagueness. In this approach, sets are defined by partial membership in contrast to crisp membership used in the classical definition of a set. Rough set theory [32] expresses vagueness not by means of membership but by employing the boundary region of a set. If the boundary region of a set is empty it means that a particular set is crisp, otherwise the set is rough (inexact). The non-empty boundary region of the set means that our knowledge about the set is not sufficient to define the set precisely. In this paper, some consequences on understanding of vague concepts caused by inductive extensions of approximation spaces and adaptive concept learning are presented. A discussion on vagueness in the context of fuzzy sets and rough sets can be found in [40].

Initially, the approximation spaces were introduced for decision tables (samples of objects). The assumption was made that the partial information about objects is given by values of attributes and on the basis of such information about objects the approximations of subsets of objects form the universe restricted to sample have been defined [32]. Starting, at least, from the early 90s, many researchers have been using the rough set approach for constructing classification algorithms (classifiers) defined over extensions of samples. This is based on the assumption that available information about concepts is partial. In recent years, there have been attempts based on approximation spaces and operations on approximation spaces for developing an approach to approximation of concepts over the extensions of samples (see, e.g., [48,50,51,56]). In this paper, we follow this approach and we show that the basic operations related to approximation of concepts on extension of samples are inductive extensions of approximation spaces. For illustration of the approach we use approximation spaces defined in [47]. Among the basic components of approximation spaces are neighborhoods of objects defined by the available information about objects and rough inclusion functions between sets of objects. Observe that searching for relevant (for approximation of concepts) extensions of approximation spaces requires tuning many more parameters than in the case of approximation of concepts on samples. The important conclusion from our considerations is that the inductive extensions used in constructing of algorithms (classifiers) are defined by arguments "for" and "against" of concepts. Each argument is defined by a tuple consisting of a degree of inclusion of objects into a pattern and a degree of inclusion of the pattern into the concepts. Patterns in the case of rule-based classifiers can be interpreted as the left hand sides of decision rules. The arguments are discovered from available data and can be treated as the basic information granules used in the concept approximation process. For any new object, it is possible to check the satisfiability of arguments and select arguments satisfied to a satisfactory degree. Such selected arguments are fused by conflict resolution strategies for obtaining the classification decision. Searching for relevant approximation spaces in the case of approximations over extensions of samples requires discovery of many parameters and patterns including selection of relevant attributes defining information about objects, discovery of relevant patterns for approximated concepts, selection of measures (similarity or closeness) of objects into discovered patters for concepts, structure and parameters of conflict resolution strategy. This causes that in the case of more compound concepts the searching process becomes infeasible (see, e.g., [6,63]). We propose to use as hints in the searching for relevant approximation spaces for compound concepts an additional domain knowledge making it possible to approximate such concepts. This additional knowledge is represented by a concept ontology [3,4,5,26,27,28,45,46,48,49,57] including concepts expressed in natural language and some dependencies between them. We assume that the ontology of concept has a hierarchical structure. Moreover, we assume that for each concept from ontology there is given a labelled set of examples of objects. The labels show the membership for objects relative to the approximated concepts. The aim is to discover the relevant conditional attributes for concepts on different levels of a hierarchy. Such attributes can be constructed using the so-called production rules, productions, and approximate reasoning schemes (AR schemes, for short) discovered from data (see, e.g. [3,4,5,26,27,28,45,46,48,49,57])). The searching for relevant arguments "for" and "against" for more compound concepts can be simplified because using domain knowledge.

Notice, that the searching process for relevant approximation spaces is driven by some selected quality measures. While in some learning problems such measures can be selected in a relatively easy way and remain unchanged during learning in other learning processes they can be only approximated on the basis of a partial information about such measures, e.g., received as the result of interaction with the environment. This case concerns, e.g., adaptive learning. We discuss the process of searching for relevant approximation spaces in different tasks of adaptive learning [1,7,12,15,21,22,24,58]. In particular, we present illustrative examples of adaptation of observation to the agent's scheme, incremental learning, reinforcement learning, and adaptive planning. Our discussion is presented in the framework of multiagent systems (MAS). The main conclusion is that the approximation of concepts in adaptive learning requires much more advanced methods. We suggest that this approach can be also based on approximation of ontology. In adaptive learning, the approximation of concepts is constructed gradually and the temporary approximations are changing dynamically in the learning process in which we are trying to achieve the approximation of the relevant quality. This, in particular, causes, e.g., boundary regions to change dynamically during the learning process in which we are attempting to find the relevant approximation of the boundary regions of approximated vague concepts. This is consistent with the requirement of the higher order vagueness [18] stating that the borderline cases of vague concepts are not crisp sets. In Sect. 5, we point out some consequences of this fact for further research on the rough set logic.

This paper is an extension and continuation of several papers (see, e.g., [3,4,5,26,27,28,44,45,46,48,49,50,56]) on approximation spaces and vague concept approximation processes. In particular, we discuss here a problem of adaptive learning of concept approximation. In this case, we are also searching for relevant approximation of the quality approximation measure. In a given step of the learning process, we have only a partial information about such a measure. On the basis of this information we construct its approximation and we use it for inducing approximation spaces relevant for concept approximation. However, in the next stages of the learning process, it may happen that after receiving new information form the environment, it is necessary to reconstruct the approximation of the quality measure and in this way we obtain a new "driving force" in searching for relevant approximation spaces during the learning process.

This paper is organized as follows. In Section 2, we discuss inductive extensions of approximation spaces. We emphasize the role of discovery of special patterns and the so called arguments in inductive extensions. In Section 3, the role of approximation spaces in hierarchical learning is presented. Section 3, outlines and approach based on approximation spaces in adaptive learning. In Sect. 5 (Conclusions), we summarize the discussion presented in the paper and we present some further research directions based on approximation spaces to approximate reasoning in multiagent systems and complex adaptive systems.

2 Approximation Spaces and Their Inductive Extensions

In [32], any approximation space is defined as a pair (U, R), where U is a universe of objects and $R \subseteq U \times U$ is an indiscernibility relation defined by an attribute set.

The lower approximation, the upper approximation and the boundary region are defined as crisp sets. It means that the higher order vagueness condition is not satisfied [18]. We will return to this issue in Section 4.

We use the definition of approximation space introduced in [47]. Any approximation space is a tuple $AS = (U, I, \nu)$, where U is the universe of objects, I is an uncertainty function, and ν is a measure of inclusion called the inclusion function, generalized in rough mereology to the rough inclusion [47,51].

In this section, we consider the problem of approximation of concepts over a universe U^*, i.e., subsets of U^*. We assume that the concepts are perceived only through some subsets of U^*, called samples. This is a typical situation in machine learning, pattern recognition, or data mining [10]. In this section we explain the rough set approach to induction of concept approximations. The approach is based on inductive extension of approximation spaces.

Now we will discuss in more detail the approach presented in [50,51]. Let $U \subseteq U^*$ be a finite sample and let $C_U = C \cap U$ for any concept $C \subseteq U^*$. Let $AS = (U, I, \nu)$ be an approximation space over the sample U. The problem we consider is how to extend the approximations of C_U defined by AS to approximation of C over U^*. We show that the problem can be described as searching for an extension

$AS^* = (U^*, I^*, \nu^*)$ of the approximation space AS relevant for approximation of C. This requires showing how to induce values of the extended inclusion function to relevant subsets of U^* that are suitable for the approximation of C. Observe that for the approximation of C, it is enough to induce the necessary values of the inclusion function ν^* without knowing the exact value of $I^*(x) \subseteq U^*$ for $x \in U^*$.

We consider an example for rule-based classifiers[1]. However, the analogous considerations for k-NN classifiers, feed-forward neural networks, and hierarchical classifiers [10] show that their construction is based on the inductive extension of inclusion function [51,44].

Usually, neighborhoods of objects in approximation spaces are defined by some formulas called patterns. Let us consider an example. Let AS^* be a given approximation space over U^* and let us consider a language L of patterns, where x denotes an object from U^*. In the case of rule-based classifiers, patterns are defined by feature value vectors. More precisely, in this case any pattern $pat(x)$ is defined by a formula $\bigwedge\{(a, a(x)) : a \in A \text{ and } v_a \in V_a\}$, where A is a given set of condition attributes [32]. An object $u \in U^*$ is satisfying $\bigwedge\{(a, a(x)) : a \in A \text{ and } v \in V_a\}$ if $a(u) = a(x)$ for any $a \in A$, i.e., if and only if x, u are A-indiscernible [32]. The set of objects satisfying $pat(x)$ in U^*, i.e., the semantics of $pat(x)$ in U^*, is denoted by $\|pat(x)\|_{U^*}$. Hence, $\|pat(x)\|_{U^*} = [x]_A$ where $[x]_A$ is the A-indiscernibility class of $x \in U^*$ [32]. By $\|pat(x)\|_U$ we denote the restriction of $\|pat(x)\|$ to $U \subseteq U^*$, i.e., the set $\|pat(x)\| \cap U$. In the considered case, we assume that any neighborhood $I(x) \subseteq U$ in AS is expressible by a pattern $pat(x)$. It means that $I(x) = \|pat(x)\|_U \subseteq U$, where $\|pat(x)\|_U$ denotes the meaning of $pat(x)$ restricted to the sample U.

We assume that for any object $x \in U^*$, only partial information about x (resulting, e.g., from a sensor measurement) represented by a pattern $pat(x) \in L$ with semantics $\|pat(x)\|_{U^*} \subseteq U^*$ defining the neighborhood of x in U^* is available. Moreover, only partial information such as $\|pat(x)\|_U$ is available about this set. In particular, relationships between information granules over U^*, e.g., $\|pat(x)\|_{U^*}$ and $\|pat(y)\|_{U^*}$, for different $x, y \in U^*$, are known, in general, only to a degree estimated by using relationships between the restrictions of these sets to the sample U, i.e., between sets $\|pat(x)\|_{U^*} \cap U$ and $\|pat(y)\|_{U^*} \cap U$.

The set $\{pat(x) : x \in U\}$ of patterns (defined by the whole set of attributes A from from an approximation space AS) is usually not relevant for approximation of the concept $C \subseteq U^*$. Such patterns can be too specific or not general enough, and can directly be applied only to a very limited number of new sample elements. For example, for a new object $x \in U^* \setminus U$ the set $\|pat(x)\|_U$ can be empty.

However, by using some generalization strategies, one can induce from patterns belonging to $\{pat(x) : x \in U\}$ some new patterns that are relevant for approximation of concepts over U^*.

[1] For simplicity of reasoning we consider only binary classifiers, i.e. classifiers with two decision classes. One can easily extend the approach to the case of classifiers with more decision classes.

Usually, first we define a new set PAT of patterns, which are candidates for relevant approximation of a given concept C. A typical example of the set of such patterns used in the case of rule based classifiers can be defined by dropping some descriptors from patterns constructed over the whole set of attributes, i.e., $\{\bigwedge\{(a, a(x)) : a \in B \text{ and } v_a \in V_a\} : B \subseteq A \text{ and } x \in U\}$. Among such patterns we search for the left hand sides of decision rules.

The set $PATTERNS(AS, L, C)$ can be selected from PAT using some quality measures evaluated on meanings (semantics) of patterns from this set restricted to the sample U. Often such measures are based on the numbers of examples from the concept C_U and its complement that support (satisfy) a given pattern. For example, if the confidence coefficient

$$\frac{card(\|pat\|_U \cap C_U)}{card(\|pat\|_U)}, \tag{1}$$

where $pat \in PAT$, is at least equal to a given threshold and the support

$$\frac{card(\|pat\|_U \cap C_U)}{card(U)}, \tag{2}$$

is also at least equal to a given threshold than we select pat as a member of $PATTERNS(AS, L, C)$.

Next, on the basis of some properties of sets definable by patterns from $PATTERNS(AS, L, C)$ over U, we induce approximate values of the inclusion function $\nu^*(X, C)$ on subsets of $X \subseteq U^*$ definable by any such pattern and the concept C. For example, we assume that the value of the confidence coefficient is not changing significantly if we move from U to U^*, i.e.,

$$\frac{card(\|pat\|_U \cap C_U)}{card(\|pat\|_U)} \approx \frac{card(\|pat\|_{U^*} \cap C)}{card(\|pat\|_{U^*})}, \tag{3}$$

Next, we induce the value of ν^* on pairs (X, Y) where $X \subseteq U^*$ is definable by a pattern from $\{pat(x) : x \in U^*\}$ and $Y \subseteq U^*$ is definable by a pattern from $PATTERNS(AS, L, C)$. For example, if $pat(x) = \bigwedge\{(a, a(x)) : a \in A \text{ and } v_a \in V_a\}$ and pat is obtained from $pat(x)$ by dropping some conjuncts then $\nu^*(\|pat(x)\|_{U^*}, \|pat\|)_{U^*}) = 1$ because $\|pat(x)\|_{U^*} \subseteq \|pat\|_{U^*}$. In a more general case, one can estimate the degree of inclusion of $\|pat(x)\|_{U^*}$ into $\|pat\|_{U^*}$ using some similarity degrees defined between formulas from PAT and $PATTERNS(AS, L, C)$. For example, one can assume that the values of attributes on x which occur in pat are not necessarily the same but similar. Certainly, such a similarity should be also defined or learned from data.

Finally, for any object $x \in U^* \setminus U$ we induce the degree $\nu^*(\|pat(x)\|_{U^*}, C)$ applying a conflict resolution strategy $Conflict\_res$ (e.g., a voting strategy) to the family of tuples:

$$\{(\nu^*(\|pat(x)\|_{U^*}, \|pat\|_{U^*}), pat, \nu^*(\|pat\|_{U^*}, C)) : pat \in PATTERNS(AS, L, C)\}. \tag{4}$$

Let us observe that conflicts can occur due to inductive reasoning in estimation of values of ν^*. For some $x \in U^*$ and $pat, pat' \in PATTERNS(AS, L, C)$ the values $\nu^*(\|pat(x)\|_{U^*}, \|pat\|_{U^*})$, $\nu^*(\|pat(x)\|_{U^*}, \|pat'\|_{U^*})$ can be both large (i.e., close to 1) and at the same time the value $\nu^*(\|pat\|_{U^*}, C)$ can be small (i.e., close to 0) and the value of $\nu^*(\|pat'\|_{U^*}, C)$ can be large.

Values of the inclusion function for the remaining subsets of U^* can be chosen in any way – they do not have any impact on the approximations of C. Moreover, observe that for the approximation of C we do not need to know the exact values of uncertainty function I^* – it is enough to induce the values of the inclusion function ν^*. The defined extension ν^* of ν to some subsets of U^* makes it possible to define an approximation of the concept C in a new approximation space AS^*.

To reduce the number of conditions from (4) one can use the so called arguments "for" and "against" discussed, e.g., in [49].

Any C-argument, where $C \subseteq U^*$ is a concept is a triple

$$(\epsilon, pat, \epsilon') \tag{5}$$

where $\epsilon, \epsilon' \in [0, 1]$ are degrees and pat is a pattern from $PATTERNS(AS, L, C)$.

The argument $arg = (\varepsilon, pat, \varepsilon')$ is satisfied by a given object $x \in U^*$, in symbols $x \models_C arg$, if and only if the following conditions are satisfied:

$$\begin{aligned} &\nu^*(\|pat(x)\|_{U^*}, \|pat\|_{U^*}) \geq \varepsilon; \\ &\nu^*(\|pat\|_{U^*}, C) \geq \varepsilon'. \end{aligned} \tag{6}$$

The idea of C-arguments is illustrated in Figure 1.

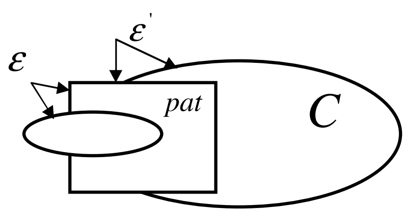

Fig. 1. C-argument

Instead of all conditions from (4) only some arguments "for" and "against" C are selected and the conflict resolution strategy is based on them. For any new object the strategy resolves conflicts between selected arguments "for" and "against" C which are satisfied by the object.

The very simple strategy for selection of arguments is the following one. The C-argument $arg = (\varepsilon, pat, \varepsilon')$ is called the argument "for" C if $\varepsilon \geq t$ and $\varepsilon' \geq t'$, where $t, t' > 0.5$ are given thresholds. The argument arg is "against" C, if this argument is the argument for the complement of C, i.e., for $U^* \setminus C$. However, in general, this may be not relevant method and the relevant arguments should be selected on the basis of more advanced quality measures. They can take into account, e.g., the support of patterns in arguments (see Section 4.1), their coverage, independence from other arguments, or relevance in searching for arguments used for approximation of more compound concepts in hierarchical learning.

One can define the lower approximation and upper approximation of the concept $C \subseteq U^*$ in the approximation space AS^* by

$$LOW(AS^*, C) = \{x \in U^* : \nu^*(I^*(x), C) = 1\}, \quad (7)$$
$$UPP(AS^*, C) = \{x \in U^* : \nu^*(I^*(x), C) > 0\}.$$

From the definition, in the case of standard rough inclusion [48], we have:

$$LOW(AS^*, C) \cap U \subseteq C \cap U \subseteq UPP(AS^*, C) \cap U. \quad (8)$$

However, in general the following equalities do not hold:

$$LOW(AS, C \cap U) = LOW(AS^*, C) \cap U, \quad (9)$$
$$UPP(AS, C \cap U) = UPP(AS^*, C) \cap U.$$

One can check that in the case of standard rough inclusion [48] we have:

$$LOW(AS, C \cap U) \supseteq LOW(AS^*, C) \cap U, \quad (10)$$
$$UPP(AS, C \cap U) \subseteq UPP(AS^*, C) \cap U.$$

Following the minimal length principle [41,42,52] some parameters of the induced approximation spaces are tuned to obtain a proper balance between the description length of the classifier and its consistency degree. The consistency degree on a given sample U of data can be represented by degrees to which the sets defined in equalities (9) are close. The description length is measured by description complexity of the classifier representation. Among parameters which are tuned are attribute sets used in the classifier construction, degrees of inclusion of patterns defined by objects to the left hand sides of decision rules, degrees of inclusion of patterns representing the left hand sides of decision rules in the decision classes, the specificity or support of these patterns, parameters of the conflict resolution strategy (e.g., set of arguments and parameters of arguments).

We can summarize our considerations in this section as follows. The inductive extensions of approximation spaces are basic operations on approximation spaces in searching for relevant approximation spaces for concept approximation. The approximation of concepts over U^* is based on searching for relevant approximation spaces AS^* in the set of approximation spaces defined by inductive extensions of a given approximation space AS. For any object $x \in U^* \setminus U$, the value $\nu^*(I^*(x), C)$ of the induced inclusion function ν^* is defined by conflict resolution strategy from collected arguments *for* classifying x to C and from collected arguments *against* classifying x to C.

3 Approximation Spaces in Hierarchical Learning

The methodology for approximation spaces extension presented in Section 2 is widely used for construction of rule based classifiers. However, this methodology cannot be directly used for concepts that are compound because of problems with inducing of the relevant set $PATTERNS(AS, L, C)$ of patterns. For such compound concepts, hierarchical learning methods have been developed (see, e.g., [2,3,4,5,26,27,28,45,46,48,49,57]).

We assume that domain knowledge is available about concepts. There is given a hierarchy of concepts and dependencies between them creating the so-called *concept ontology*. Only partial information is available about concepts in the hierarchy.

For concepts from the lowest level of hierarchy, decision tables with condition attributes representing sensory measurements are given. Classifiers for these concepts are induced (constructed) from such decision tables. Assuming that classifiers have been induced for concepts from some level l of the hierarchy, we are aiming at inducing classifiers for concepts on the next $l+1$ level of the hierarchy. It is assumed that for concepts on higher levels there are given samples of objects with information about their membership values relative to the concepts. The relevant patterns for approximation of concepts from the $l+1$ level are discovered using (i) these decision tables, (ii) information about dependencies linking concepts from the level $l+1$ with concepts from the level l, and (iii) patterns discovered for approximation of concepts from the level l of the hierarchy. Such patterns define condition attributes (e.g., by the characteristic functions of patterns) in decision tables. Next, using the condition attributes approximation of concepts are induced. In this way, also, the neighborhoods for objects on the level $l+1$ are defined. Observe also that the structure of objects on the higher level $l+1$ is defined by means of their parts from the level l. In this section, for simplicity of reasoning, we assume that on each level the same objects are considered. To this end, we also assume that rough inclusion functions from approximation spaces are standard rough inclusion functions [48].

Now we outline a method of construction of patterns used for approximation of concepts from a given level of concept hierarchy by patterns used for approximation of concepts belonging to the lower level of the hierarchy.

This approach has been elaborated in a number of papers cited above, in particular in [49]. Assume that a concept C belongs to a level $l+1$ of the hierarchy. We outline the idea of searching for sets $PATTERNS(AS, L, C)$ of patterns for a concept C, where AS is an approximation space discovered for approximation of the concept C and L is a language in which discovered patterns are expressed.

To illustrate this idea, let us consider and example of a dependency for a concept C from domain knwoledge:

$$\text{if } C_1 \text{ and } C_2 \text{ then } C, \tag{11}$$

where C_1, C_2, C are vague concepts. Analogously, let us consider a dependency for the complement of C:

$$\text{if } C_1' \text{ and } C_2' \text{ then } \neg C. \tag{12}$$

In general, we should consider a set with many dependencies with different concepts on the right hand sides of dependencies (creating, e.g., a partition of the universe) and in the process of generating arguments "for" and "against" a selected concept C are involved other vague dependencies from the given set. Let us recall that such a set of concepts and dependencies between them is specified in a given domain knowledge.

To approximate the target concept C, relevant patterns for C and $\neg C$ should be derived. The main idea is presented in Figure 2 and Figure 3. We assume that for any considered concept and for each pattern selected for this concept a degree of its inclusion into the concept can be estimated.

In Figure 2 it is shown that for patterns pat_1, pat_2 (e.g., left hand sides of decision rules in case of a rule based classifiers) for (or against) C_1 and C_2 and their inclusion degrees ϵ_1 and ϵ_2 into C_1 and C_2, respectively, it is constructed a pattern *pat* for (or against) C together with estimation of its inclusion degree ϵ to the concept C.

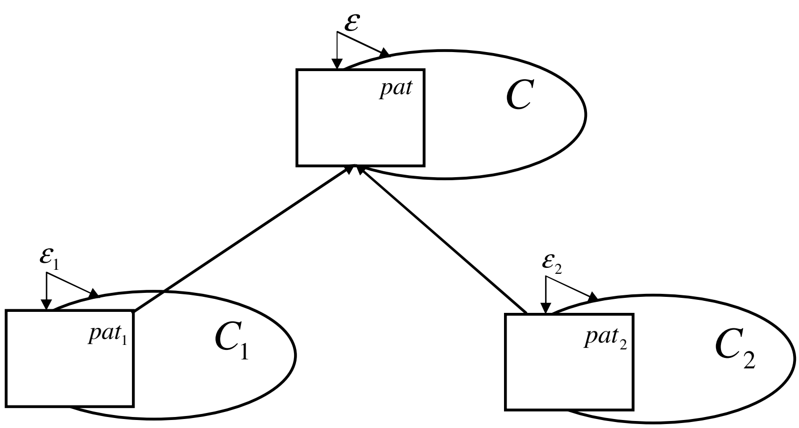

Fig. 2. An illustration of pattern construction

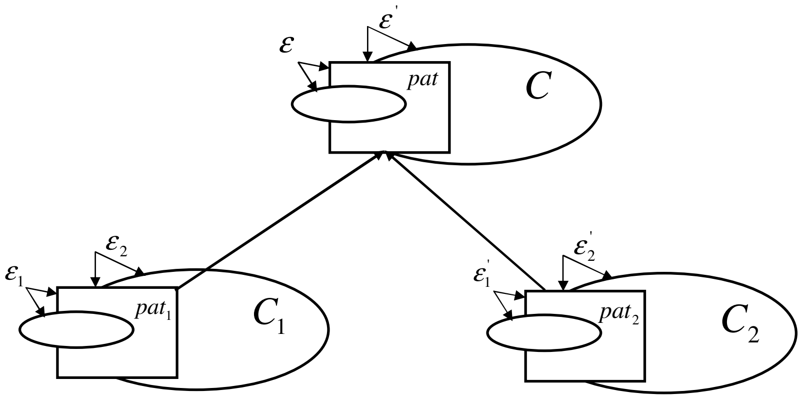

Fig. 3. An illustration of production rule

Figure 3 represents a construction of the target argument $(\epsilon, pat, \epsilon')$ for C from arguments $(\epsilon_1, pat_1, \epsilon'_1)$ and $(\epsilon_2, pat_2, \epsilon'_2)$ for C_1 and C_2, respectively. Such a construction

$$\text{if } (\epsilon_1, pat_1, \epsilon'_1) \text{ and } (\epsilon_2, pat_2, \epsilon'_2) \text{ then } (\epsilon, pat, \epsilon') \tag{13}$$

is called a *production rule* for the dependency (11). This production rule is true at a given object x if and only if the following implication holds:

$$\text{if } x \models_{C_1} (\epsilon_1, pat_1, \epsilon'_1) \text{ and } x \models_{C_2} (\epsilon_2, pat_2, \epsilon'_2) \text{ then } x \models_C (\epsilon, pat, \epsilon'). \tag{14}$$

Certainly, it is necessary to search for production rules of the high quality (with respect to some measures) making it possible to construct "strong" arguments in the conclusion of the production from "strong" arguments in the premisses of the production rule. The quality of arguments is defined by means of relevant degrees of inclusion in these arguments and properties of patterns (such as support or description length).

The quality of arguments for concepts from the level $l + 1$ can be estimated on the basis properties of arguments for the concepts from the level l from which these arguments have been constructed. In this estimation are used decision tables delivered by domain experts. Such decision tables consist of objects with decision values equal to the membership degrees of objects relative to the concept or to its complement. In searching for productions of high quality, we use operations called *constrained sums* (see, e.g., [55]). Using these operations there are performed joins of information systems representing patterns appearing in arguments from the premise of production. The join is parameterized by constraints helping by tuning these parameters to filter the relevant objects from composition of patterns used for constructing a pattern for the concept C on the level $l + 1$ for the argument in the conclusion of the production rule. Moreover, the production rules may be composed into the so called approximation reasoning schemes (AR schemes, for short). This makes it possible to generate patterns for approximation of concepts on the higher level of the hierarchy (see, e.g., [2,3,4,5,26,27,28,46,49]). In this way one can induce gradually for any concept C in the hierarchy a relevant set of arguments (based on the relevant set of patterns $PATTERNS(AS, L, C)$ of patterns; see Section 3) for approximation of C.

We have recognized that for a given concept $C \subseteq U^*$ and any object $x \in U^*$, instead of crisp decision about the relationship of $I^*(x)$ and C, we can gather some arguments *for* and *against* it only. Next, it is necessary to induce from such arguments the value $\nu^*(I(x), C)$ using some strategies making it possible to resolve conflicts between those arguments [10,48]. Usually some general principles are used such as the minimal length principle [10] in searching for algorithms computing an extension $\nu^*(I(x), C)$. However, often the approximated concept over $U^* \setminus U$ is too compound to be induced directly from $\nu(I(x), C)$. This is the reason that the existing learning methods are not satisfactory for inducing high quality concept approximations in case of complex concepts [63]. There have been several attempts trying to omit this drawback. In this section we have discussed the approach based on hierarchical (layered) learning [57].

There are some other issues which should be discussed in approximation of compound vague concepts. Among them are issues related to adaptive learning and construction or reconstruction of approximation spaces in interaction with environments. In the following section, we consider an agent learning some concepts. This agent is learning the concepts in interaction with the environments. Different types of interaction are defining different types of adaptive learning processes. In particular one can distinguish incremental learning [13,23,61,65], reinforcement learning [9,14,17,34,39,60], competitive or cooperative learning [15]. There are several issues, important for adaptive learning that should be mentioned. For example, the compound target concept which we attempt to learn can gradually change over time and this concept drift is a natural extension for incremental learning systems toward adaptive systems. In adaptive learning it is important not only what we learn but also how we learn, how we measure changes in a distributed environment and induce from them adaptive changes of constructed concept approximations. The adaptive learning for autonomous systems became a challenge for machine learning, robotics, complex systems, and multiagent systems. It is becoming also a very attractive research area for the rough set approach. Some of these issues will be discussed in the following section.

4 Approximation Spaces in Adaptive Learning

There are different interpretations of the terms *adaptive learning* and *adaptive systems* (see, e.g., [1,7,12,15,21,22,24,58]). We mean by *adaptive* a system that learns to change with its environment. Our understanding is closest to the spirit of what appears in [7,12]. In complex adaptive systems (CAS), agents scan their environment and develop a schema for action. Such a schema defines interactions with agents surrounding it together with information and resources flow externally [7]. In this section, we concentrate only on some aspects of adaptive learning. The other issues of adaptive learning in MAS and CAS will be discussed elsewhere.

In particular, we would like to discuss the role of approximation spaces in adaptive learning.

In this paper, we consider the following exemplary situation. There is an agent *ag* interacting with another agent called the environment (ENV). Interactions are represented by actions [11,62] performed by agents. These actions are changing values of some sensory attributes of agents. The agent *ag* is equipped with ontology of vague concepts consisting of vague concepts and dependencies between them.

There are three main tasks of the agent *ag*: (i) adaptation of observation to the agent's scheme, (ii) adaptive learning of the approximations of vague concepts, and (iii) preserving constraints (e.g., expressed by dependencies between concepts).

Through adaptation of observation to the agent's scheme agent becomes more robust and can handle more variability [7].

Approximation of vague concepts by the agent ag requires development of searching methods for relevant approximation spaces which create the basis for approximation of concepts. Observe that the approximations of vague concepts are dynamically changing in adaptive learning when new knowledge about approximated concept is obtained by the agent ag. In particular, from this it follows that the boundary regions of approximated concepts are dynamically changing in adaptive learning. For each approximated concept we obtain a sequence of boundary regions rather than a single crisp boundary region. By generating this sequence we are attempting to approximate the set of borderline cases of a given vague concept. Hence, if the concept approximation problem is considered in adaptive framework the higher order postulate for vague concepts is satisfied (i.e., the set of borderline cases of any vague concept can not be crisp) [18,44,50].

The third task of the agent ag requires learning of a planning strategy. This is a strategy for predicting plans (i.e., sequences of actions) on the basis of observed changes in the satisfiability of the observed concepts from ontology. By executing plans the actual state of the system is transformed to a state satisfying the constraints. Changes in the environments can cause that the executed plans should be reconstructed dynamically by relevant adaptive strategies. In our example, actions performed by the agent ag are adjusting values of sensory attributes which are controllable by ag.

Before we will discuss the mentioned above tasks in more detail we would like to add some comments on interaction between agents.

The interactions among agents belong to the most important ingredients of computations realized by multiagent systems [21]. In particular, adaptive learning agents interact, in particular, with their environments. In this section, we will continue our discussion on adaptive learning by agents interacting with environment. Some illustrative examples of interactions which have influence on the learning process are presented.

Let us consider two agents ag and ENV representing the agent learning some concepts and the environment, respectively. By $ag_s(t)$ and $ENV_s(t)$ we denote (information about) the state of agents ag and ENV at the time t, respectively. Such an information can be represented, e.g., by a vector of attribute values A_{ag} and A_{ENV}, respectively [51]. The agent ag is computing the next state $ag_s(t+1)$ using his own transition relation $\longrightarrow_{ag}$ applied to the result of interaction of $ag_s(t)$ and $ENV_s(t)$. The result of such an interaction we denote by $ag_s(t) \oplus_{ENV} ENV_s(t)$ where $\oplus_{ENV}$ is an operation of interaction of ENV on the state of ag. Hence, the following condition holds:

$$ag_s(t) \oplus_{ENV} ENV_s(t) \longrightarrow_{ag} ag_s(t+1). \tag{15}$$

Analogously, we obtain the following transition for environment states:

$$ag_s(t) \oplus_{ag} ENV_s(t) \longrightarrow_{ENV} ENV_s(t+1). \tag{16}$$

In our examples, we will concentrate on two examples of interactions. In the first example related to incremental learning (see, e.g., [13,23,61,65]), we assume that $ag_s(t) \oplus_{ENV} ENV_s(t)$ is obtained by extending of $ag_s(t)$ by a new information about some new sample of objects labelled by decisions. The structure of $ag_s(t)$ is much more compound than in non-incremental learning. This will be discussed in one of the following section together with some aspects of adaptation in incremental learning. These aspects are related to searching for relevant approximation spaces. In the discussed case, we also assume that $ag_s(t) \oplus_{ag} ENV_s(t) = ENV_s(t)$, i.e., there is no interaction of the agent ag on the environment. In our second example, the agent ag can change the state of ENV by performing some actions or plans which change the state of the environment.

4.1 Adaptation of Observation to the Agent's Scheme

In this section, we present two illustrative examples of adaptation of observation to the agent's scheme. In the consequence of such an adaptation, the agent's scheme becomes more robust relative to observations.

In the first example, we consider instead of patterns $pat(x)$ (see Section 2) mode general patterns which are obtained by granulation of such patterns using a similarity relation τ. Assuming that the object description $pat(x)$ is defined by $\bigwedge\{(a, a(x)) : a \in A \text{ and } v_a \in V_a\}$ one can define such a similarity τ on description of objects, e.g., by a composition of similarity relations on attribute value sets (see, e.g., [20,25,47])[2]. Then instead of patterns $pat(x)$ we obtain patterns $pat_\tau(x)$ with the semantics defined by $\|pat_\tau(x)\|_{U^*} = \{y \in U^* : pat(x)\tau pat(y)\}$. The definition of satisfiability of arguments (6) changes as follows

$$\begin{aligned} &\nu^*(\|pat_\tau(x)\|_{U^*}, \|pat\|_{U^*}) \geq \varepsilon; \\ &\nu^*(\|pat\|_{U^*}, C) \geq \varepsilon'. \end{aligned} \tag{17}$$

Observe, that $\|pat_\tau(x)\|_{U^*}$ is usually supported by many more objects than $\|pat(x)\|_{U^*}$. Hence, if it is possible to tune the parameters of τ in such a way that the first condition in (17) is satisfied for sufficiently large ε than the obtained argument is much more robust than the previous one, i.e., it is satisfied by much more objects than the previous one $pat(x)$ and at the same time the requirement related to the degrees of inclusion is preserved.

Our second example concerns construction of more robust production rules and productions (sets of production rules corresponding to the same dependency between vague concepts) (see Figure 4). Patterns in such productions represent different layers of vague concepts and are determined by the linguistic values of membership such as *small, medium, high* (see, e.g., [5]). These more general patterns are constructed using information granulation [51]. Let us consider a simple example of information granulation. Observe that the definition of the satisfiability of arguments given by (6) is not unique. One can consider the decision table (U, A, d), where A is a given set of condition attributes [32] and the decision d is the characteristic function of the set $Y_\varepsilon(pat) = \{y \in U : \nu(\|pat(y)\|_U, \|pat\|_U)) \geq \varepsilon\}$. From this decision table can be induced the classifier $Class(pat)$ for the concept $Y^*_\varepsilon(pat) = \{y \in U^* : \nu^*(\|pat(y)\|_{U^*}, \|pat\|_{U^*})) \geq \varepsilon\}$.

[2] Note, that the similarity relation τ has usually many parameters which should be tuned in searching for relevant similarity relations.

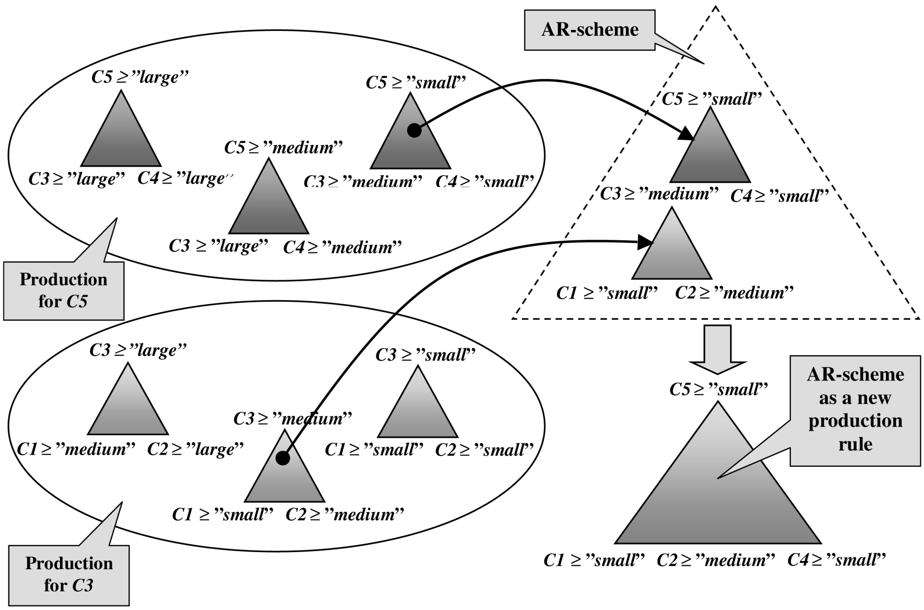

Fig. 4. An illustration of production and AR scheme

Any object $x \in U^*$ is satisfying the C-argument (5) if and only if the following condition is satisfied:

$$\nu^*(Y_\varepsilon^*(pat), C) \geq \varepsilon'. \tag{18}$$

The satisfiability of (18) is estimated by checking if the following condition holds on the sample U:

$$\nu(Y_\varepsilon(pat), C \cap U) \geq \varepsilon'. \tag{19}$$

We select only the arguments $(\varepsilon, pat, \varepsilon')$ with the maximal ε' satisfying (19) for given ε and pat.

Assume that $0 = \varepsilon_0 < \ldots < \varepsilon_{i-1} < \varepsilon_i < \ldots < \varepsilon_n = 1$. For any $i = 1, \ldots, n$ we granulate a family of sets

$$\{Y_\varepsilon^*(pat) : pat \in PATTERNS(AS, L, C) \text{ and } \nu^*(Y_\varepsilon^*(pat), C) \in [\varepsilon_{i-1}, \varepsilon_i)\} \tag{20}$$

into one set $Y_\varepsilon^*(\varepsilon_{i-1}, \varepsilon_i)$. Each set $Y_\varepsilon^*(\varepsilon_{i-1}, \varepsilon_i)$ is defined by an induced classifier $Class_\varepsilon(\varepsilon_{i-1}, \varepsilon_i)$. The classifiers are induced, in an analogous way as before, by constructing a decision table over a sample $U \subseteq U^*$. In this way we obtain a family of classifiers $\{Class_\varepsilon(\varepsilon_{i-1}, \varepsilon_i)\}_{i=1,\ldots,n}$.

The sequence $0 = \varepsilon_0 < \ldots < \varepsilon_{i-1} < \varepsilon_i < \ldots < \varepsilon_n = 1$ should be discovered in such a way that the classifiers $Class_\varepsilon(\varepsilon_{i-1}, \varepsilon_i)$ correspond to different layers of the concept C with linguistic values of membership. One of the method in searching for such sequence can be based on analysis of a histogram. This histogram represents a function $f(I)$ where $I \in \mathcal{J}$, $\mathcal{J}$ is a given uniform partition of the interval $[0,1]$, and $f(I)$ is the number of patterns from $\{Y^*_\varepsilon(pat) : pat \in PATTERNS(AS,L,C)\}$ with the inclusion degree into C from $I \subseteq [0,1]$.

4.2 Adaptation and Incremental Learning

In this section, we outline a searching process for relevant approximation spaces in incremental learning. Let us consider an example of incremental concept approximation scheme Sch (see Figure 5). By $Inf(C)$ and $Inf'(C)$ we denote

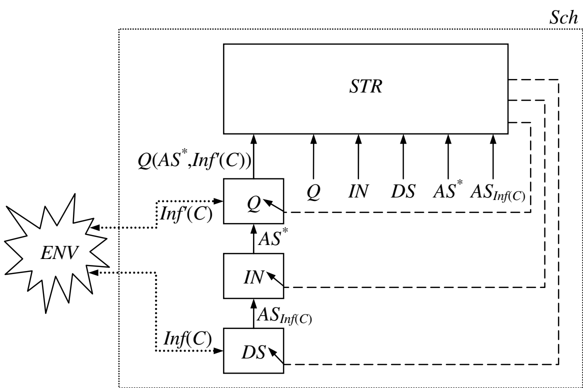

Fig. 5. An example of incremental concept approximation scheme

a partial information about the approximated concept (e.g., decision table for C or training sample) in different moments t and $t+1$ of time, respectively[3]. ENV denotes an environment, DS is an operation constructing an approximation space $AS_{Inf(C)}$ from a given sample $Inf(C)$, i.e., a decision table. IN is an inductive extension operation (see Section 2) transforming the approximation space $AS_{Inf(C)}$ into an approximation space AS^* used for approximation of the concept C; Q denotes a quality measure for the induced approximation space AS^* on a new sample $Inf'(C)$, i.e., an extension of the decision table $Inf(C)$. For example, the value $Q(AS^*, Inf'(C))$ can be taken as a ratio of the number of objects from $Inf'(C)$ that are classified correctly (relative to the decision values from $Inf'(C)$) by the classification algorithm (classifier) for C defined by AS^* (see Section 2) to the number of all objects in $Inf'(C)$.

The double-ended arrows leading into and out of ENV illustrate an interaction between agent ag and the environment ENV. In the case of a simple incremental learning strategy only samples of C are communicated by ENV to ag. More compound interactions between ag and ENV will be discuss later. They can be related to reaction from ENV on predicted by ag decisions (actions, plans) (see, e.g., award and penalty policies in reinforcement strategies [9,17,14,34,39,60]).

[3] For simplicity, in Figure 5 we do not present time constraints.

STR is a strategy that adaptively changes the approximation of C by modifying the quality measure Q, the operation of inductive extension IN, and the operation DS of constructing the approximation space $AS_{Inf(C)}$ from the sample $Inf(C)$. Dotted lines outgoing from the box labelled by SRT in Figure 5 are illustrating that the strategy STR after receiving the actual values of input its parameters is changing them (e.g., in the next moment of time). To make Figure 5 more readable the dotted lines are pointing to only one occurrence of each parameter of STR but we assume that its occurrences on the input for STR are modified too.

In the simple incremental learning strategy the quality measure is fixed. The aim of the strategy STR is to optimize the value of Q in the learning process. This means that in the learning process we would like to reach as soon as possible an approximation space which will guarantee the quality of classification measured by Q to be almost optimal. Still, we do not know how to control by STR this optimization. For example, should this strategy be more like the annealing strategy [19], then it is possible to perform more random choices at the beginning of the learning process and next be more "frozen" to guarantee the high convergence speed of the learning process to (semi-)optimal approximation space. In the case of more compound interactions between ag and ENV, e.g., in reinforcement learning, the quality measure Q should be learned using, e.g., awards or penalties received as the results of such interactions. This means that together with searching for an approximation space for the concept it is necessary to search for an approximation space over which the relevant quality measure can be approximated with high quality.

The scheme Sch describes an adaptive strategy ST modifying the induced approximation space AS^* with respect to the changing information about the concept C. To explain this in more detail, let us first assume that a procedure $new_C(ENV, u)$ is given returning from the environment ENV and current information u about the concept C a new piece of information about this concept (e.g., an extension of a sample u of C). In particular, $Inf^{(0)}(C) = new_C(ENV, \emptyset)$ and $Inf^{(k+1)}(C) = new_C(ENV, Inf^{(k)}(C))$ for $k = 0, \ldots$. In Figure 5 $Inf'(C) = Inf^{(1)}(C)$. Next, assuming that operations $Q^{(0)} = Q$, $DS^{(0)} = DS$, $IN^{(0)} = IN$ are given, we define $Q^{(k+1)}$, $DS^{(k+1)}$, $IN^{(k+1)}$, $AS^{(k+1)}_{(Inf^{(k+1)}(C)}$, and $AS^{*(k+1)}$ for $k = 0, \ldots$, by

$$
\begin{aligned}
&(Q^{(k+1)}, DS^{(k+1)}, IN^{(k+1)}) = \qquad (21)\\
&= STR(Q^{(k)}(AS^{*(k)}, Inf^{(k+1)}(C)), Q^{(k)}, IN^{(k)}, DS^{(k)}, AS^{*(k)}, AS^{(k)}_{Inf^{(k)}(C)})\\
&AS^{(k+1)}_{Inf^{(k+1)}(C)} = DS^{(k+1)}(Inf^{(k+1)}(C)); \quad AS^{*(k+1)} = IN^{(k+1)}(AS^{(k+1)}_{Inf^{(k+1)}(C)}).
\end{aligned}
$$

One can see that the concept of approximation space considered so far should be substituted by a more complex one represented by the scheme Sch making it possible to generate a sequence of approximation spaces $AS^{*(k)}$ for $k = 1, \ldots$ derived in an adaptive process of approximation of the concept C. One can also treat the scheme Sch as a complex information granule [48].

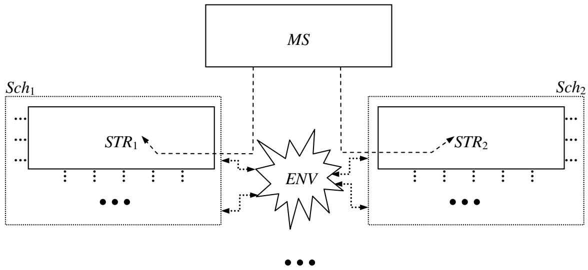

Fig. 6. An example of metastrategy in adaptive concept approximation

One can easily derive more complex adaptive schemes with metastrategies that make it possible to modify also strategies. In Figure 6 there is presented an idea of a scheme where a metastrategy MS can change adaptively also strategies STR_i in schemes Sch_i for $i = 1, \ldots, n$ where n is the number of schemes. The metastrategy MS can be, e.g., a fusion strategy for classifiers corresponding to different regions of the concept C.

4.3 Adaptation in Reinforcement Learning

In reinforcement learning [9,14,17,34,39,56,60], the main task is to learn the approximation of the function $Q(s, a)$, where s, a denotes a global state of the system and an action performed by an agent ag and, respectively and the real value of $Q(s, a)$ describes the reward for executing the action a in the state s. In approximation of the function $Q(s, a)$ probabilistic models are used. However, for compound real-life problems it may be hard to build such models for such a compound concept as $Q(s, a)$ [63]. In this section, we would like to suggest another approach to approximation of $Q(s, a)$ based on ontology approximation. The approach is based on the assumption that in a dialog with experts an additional knowledge can be acquired making it possible to create a ranking of values $Q(s, a)$ for different actions a in a given state s. We expect that in the explanation given by expert about possible values of $Q(s, a)$ are used concepts from a special ontology of concepts. Next, using this ontology one can follow hierarchical learning methods (see Section 3 and [2,3,4,5,26,27,28,45,46,48,49,57])) to learn approximations of concepts from ontology. Such concepts can have temporal character too. This means that the ranking of actions may depend not only on the actual action and the state but also on actions performed in the past and changes caused by these actions.

4.4 Adaptation and Planning

A more compound scheme than what was considered in the previous section can be obtained by considering strategies based on cooperation among the schemes for obtaining concept approximations of high quality. In Figure 7 an adaptive scheme for plan modification is presented. $PLAN$ is modified by a metastrategy MS that adaptively changes strategies in schemes Sch_i where $i = 1, \ldots, n$. This

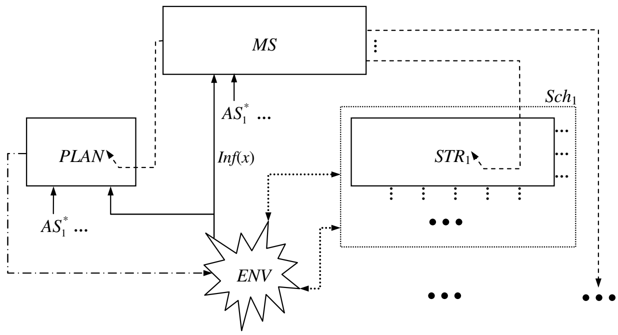

Fig. 7. An example of adaptive plan scheme

is performed on the basis of the derived approximation spaces AS_i^* induced for concepts that are guards (preconditions) of actions in plans and on the basis of information $Inf(x)$ about the state x of the environment ENV. The generated approximation spaces together with the plan structure are adaptively adjusted to make it possible to achieve plan goals.

The discussed example is showing that the context in which sequences of approximation spaces are generated can have complex structure represented by relevant adaptive schemes. The main goal of the agent *ag* in adaptive planning is to search for approximation of the optimal trajectory of states making it possible for the agent *ag* to achieve the goal, e.g., to keep as invariants some dependencies between vague concepts. Observe, that searching in adaptive learning for such a trajectory approximation should be performed together with adaptive learning of many other vague concepts which should be approximated, e.g., preconditions for actions, meta actions or plans.

One of the very important issue in adaptive learning is approximation of compound concepts used in reasoning about changes observed in the environment. The agent *ag* interacting with the environment ENV is recording changes in the satisfiability of concepts from her/his ontology. These changes should be expressed by relevant concepts (features) which are next used for construction of preconditions of actions (or plans) performed by the agent *ag*. In real-life problems these preconditions are compound concepts. Hence, to approximate such concepts we suggest to use an additional ontology of changes which can be acquired in a dialog with experts. All concepts from the ontology create a hierarchical structure. In this ontology relevant concepts characterizing changes in the satisfiability of concepts from the original ontology are included together with other simpler concepts from which they can be derived. We assume that such an ontology can be acquired in a dialog with experts. Concepts from this ontology are included in the expert explanations consisting of justifications why in some exemplary situations it is necessary to perform some particular actions in a particular order. Next, by approximation of the new ontology (see Section 3 and [2,3,4,5,26,27,28,45,46,48,49,57]) we obtain the approximation of the mentioned above compound concepts relevant for describing changes. This methodology can be used not only for predicting the relevant actions, meta actions or plans but also for the plan reconstruction. In our current projects we are developing the methodology for adaptive planning based on ontology approximation.

5 Conclusions

In the paper, we have discussed some problems of adaptive approximation of concepts by agents interacting with environments. These are the fundamental problems in synthesis of intelligent systems. Along this line important research directions perspective arise.

In particular, this paper realizes a step toward developing methods for adaptive maintenance of constraints specified by vague dependencies. Notice that there is a very important problem related to such a maintenance which should be investigated further, i.e., approximation of vague dependencies. The approach to this problem based on construction of arguments "for" and "against" for concepts from conclusions of dependencies on the basis of such arguments from premisses of dependencies will be presented in one of our next paper.

Among interesting topics for further research are also strategies for modeling of networks supporting approximate reasoning in adaptive learning. For example, AR schemes and AR networks (see, e.g., [48]) can be considered as a step toward developing such strategies. Strategies for adaptive revision of such networks and foundations for autonomous systems based on vague concepts are other examples of important issues.

In this paper also some consequences on understanding of vague concepts caused by inductive extensions of approximation spaces and adaptive concept learning have been presented. They are showing that in the learning process each temporary approximations, in particular boundary regions are crisp but they are only temporary approximations of the set of borderline cases of the vague concept. Hence, the approach we propose is consistent with the higher order vagueness principle [18].

There are some important consequences of our considerations for research on approximate reasoning about vague concepts. It is not possible to base such reasoning only on *static* models of vague concepts (i.e., approximations of given concepts [32] or membership functions [66] induced from a sample available at a given moment) and on multi-valued logics widely used for reasoning about rough sets or fuzzy sets (see, e.g., [31,36,66,69]). Instead of this there is a need for developing evolving systems of logics which in open and changing environments will make it possible to gradually acquire knowledge about approximated concepts and reason about them.

Acknowledgment

Many thanks to Professor James Peters for his incisive comments and for suggesting many helpful ways to improve this article.

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

References

1. Axaelrod, R.M.: *The Complexity of Cooperation*. Princeton University Press, Princeton, NJ (1997).
2. Bazan, J.G., Nguyen, H.S., Peters, J.F., Skowron, A., Szczuka, M.: Rough set approach to pattern extraction from classifiers. In: A. Skowron and M. Szczuka (Eds.), *Proceedings of the Workshop on Rough Sets in Knowledge Discovery and Soft Computing at ETAPS 2003* (RSKD 2003), Electronic Notes in Computer Science, **82**(4), Elsevier, Amsterdam, Netherlands, www.elsevier.nl/locate/entcs/volume82.html, April 12-13 (2003) 20–29.
3. Bazan, J.G., Peters, J.F., Skowron, A.: Behavioral pattern identification through rough set modelling. In [54] (2005) 688–697.
4. Bazan, J.G., Skowron, A.: On-line elimination of non-relevant parts of complex objects in behavioral pattern identification. In [30] (2005) 720–725.
5. Bazan, J.G., Skowron, A.: Classifiers based on approximate reasoning schemes. In B. Dunin-Keplicz, A. Jankowski A. Skowron, M. Szczuka (Eds.), *Monitoring, Security, and Rescue Tasks in Multiagent Systems MSRAS*, Advances in Soft Computing. Springer, Heidelberg (2005) 191–202.
6. Breiman, L.: Statistical modeling: The two cultures. *Statistical Science* **16**(3) (2001) 199–231.
7. Desai, A.: *Adaptive Complex Enterprices*. Communications ACM **48**(5) (2005) 32–35.
8. Dietterich, T.G.: Machine Learning Research: Four Current Directions. *AI Magazine* **18**(4) (1997) 97–136.
9. Dietterich, T.G.: Hierarchical reinforcement learning with the MAXQ value function decomposition. *Artificial Intelligence* **13** (2000) 227–303.
10. Friedman, J., Hastie, T., Tibshirani, R.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, Heidelberg (2001).
11. Ghallab, M., Nau, D., Traverso, P.: *Automated Planning: Theory and Practice*. Elsevier, Morgan Kaufmann, CA (2004).
12. Gell-Mann, M.: *The Quark and the Jaguar*. Freeman and Co., NY (1994).
13. Giraud-Carrier, Ch.: A note on the utility of incremental learning. *AI Communications* **13**(4) (2000) 215–223.
14. McGovern, A.: *Autonomous Discovery of Temporal Abstractions from Interaction with an Environment*. PhD thesis, University of Massachusetts Amherst (2002).
15. Hoen, P.J., Tuyls, K., Panait, L., Luke, S., La Poutré, J.A.: An Overview of cooperative and competitive multiagent learning. In K. Tuyls, P. J. Hoen, K. Verbeeck, S. Sen (Eds.), *Learning and Adaption in Multi-Agent Systems: First International Workshop* (LAMAS 2005), Utrecht, The Netherlands, July 25 (2005) 1–46.
16. Kautz, H., Selman, B., Jiang, Y.: A General Stochastic Approach to Solving Problems with Hard and Soft Constraints. In D. Gu, J. Du, P. Pardalos (Eds.), *The Satisfiability Problem: Theory and Applications*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **35**, American Mathematical Society (1997) 573–586.
17. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* **4** (1996) 237–285.

18. R. Keefe, R.: *Theories of Vagueness.* Cambridge Studies in Philosophy, Cambridge, UK (2000).
19. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598) (1983) 671–680.
20. Krawiec, K., Słowiński, R., Vanderpooten, D.: Lerning decision rules from similarity based rough approximations. In L. Polkowski, A. Skowron (Eds.), *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica-Verlag, Heidelberg (1998) 37–54.
21. Luck, M., McBurney, P., Shehory, O., Willmott, S.: *Agent Technology: Computing as Interaction. A Roadmap for Agent-Based Computing.* Agentlink III, the European Coordination Action for Agent-Based Computing, University of Southampton, UK (2005).
22. Marx, L.M.: Adaptive learning and iterated weak dominance. *Games and Economic Behavior* **26** (1999) 253–278.
23. Michalski, R.S., Mozetic, I., Hong, J., Lavrac, N.: The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proceedings of the 5th National Conference on Artificial Intelligence*, Philadelphia, PA. Morgan Kaufmann (1986) 1041–1047.
24. Milgrom, P., Roberts, J.: Adaptive and sophisticated learning in normal form games. *Games Economic Behavior* **3**, (1991) 82–100.
25. Nguyen, S.H.: Regularity analysis and its applications in Data Mining. In L. Polkowski, T. Y. Lin, S. Tsumoto (Eds.), *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*, Physica-Verlag, Heidelberg, Germany (2000) 289–378.
26. Nguyen, S.H., Bazan, J., Skowron, A., Nguyen, H.S.: Layered learning for concept synthesis. In J. F. Peters, A. Skowron (Eds.), *Transactions on Rough Sets I: Journal Subline, Lecture Notes in Computer Science* **3100**, Springer, Heidelberg, Germany (2004) 187–208.
27. Nguyen, T.T., Skowron, A.: Rough set approach to domain knowledge approximation. In [64] (2003) 221–228.
28. Nguyen, T.T.: Eliciting domain knowledge in handwritten digit recognition. In [30] (2005) 762–767.
29. Pal, S.K, Polkowski, L., Skowron, A. (Eds.): *Rough-Neural Computing: Techniques for Computing with Words*, Cognitive Technologies, Springer-Verlag, Heidelberg (2004).
30. Pal, S.K, Bandoyopadhay, S., Biswas, S. (Eds.). *Proceedings of the First International Conference on Pattern Recognition and Machine Intelligence* (PReMI'05), December 18-22, 2005, Indian Statistical Institute, Kolkata, Lecture Notes in Computer SCience **3776**. Springer-Verlag, Heidelberg, Germany (2005).
31. Pavelka, J.: On Fuzzy Logic I-III. *Zeit. Math Logik Grund. Math.* **25** (1979) 45–52, 119-134, 447-464.
32. Pawlak, Z.: *Rough Sets: Theoretical Aspects of Reasoning about Data.* System Theory, Knowledge Engineering and Problem Solving **9**, Kluwer Academic Publishers, Dordrecht (1991).
33. Pawlak, Z., Skowron, A.: Rough Membership Functions. In R. R. Yager, M. Fedrizzi, J. Kacprzyk (eds.), *Advances in the Dempster-Schafer Theory of Evidence*, John Wiley and Sons, New York (1994) 251–271.
34. Peters, J.F., Henry, C.: Reinforcement Learning with Approximation Spaces. *Fundamenta Informaticae* **71**(2-3) (2006) 323–349.

35. Peters, J.F., Skowron, A. (Eds.): Transactions on Rough Sets III: Journal Subline, Lecture Notes in Computer Science, vol. 3400. Springer, Heidelberg, Germany (2005).
36. Polkowski, L.: *Rough Sets: Mathematical Foundations.* Physica-Verlag, Heidelberg (2002).
37. Polkowski, L., Skowron, A.: Rough Mereology: A New Paradigm for Approximate Reasoning. *International Journal of Approximate Reasoning* **15**, (1996) 333–365.
38. Polkowski, L., Skowron, A.: Rough Mereological Calculi of Granules: A Rough Set Approach to Computation. *Computational Intelligence* **17**, (2001) 472–492.
39. Randlov, J.: *Solving Complex Problems with Reinforcement Learning.* Ph.D. Thesis, University of Copenhagen (2001).
40. Read, S.: *Thinking about Logic. An Introduction to the Philosophy of Logic.* Oxford University Press, Oxford, New York (1995).
41. Rissanen, J.: Modeling by shortes data description. Automatica **14** (1978) 465–471.
42. Rissanen, J.: Minimum-description-length principle. In: S. Kotz, N. Johnson (Eds.), Encyclopedia of Statistical Sciences, John Wiley & Sons, New York, NY (1985) 523–527.
43. Skowron, A.: Rough Sets in KDD. *16-th World Computer Congress* (IFIP'2000), Beijing, August 19-25, 2000, In Z. Shi, B. Faltings, M. Musen (Eds.), *Proceedings of Conference on Intelligent Information Processing* (IIP2000), Publishing House of Electronic Industry, Beijing (2000) 1–17.
44. Skowron, A.: Rough Sets and Vague Concepts. *Fundamenta Informaticae* **4(1-4)** (2005) 417–431.
45. Skowron, A.: Approximate reasoning in distributed environments. In N. Zhong, J. Liu (Eds.), *Intelligent Technologies for Information Analysis*, Springer, Heidelberg (2204) 433–474.
46. Skowron, A., Peters, J.F.: Rough sets: Trends and challenges. In [64] (2003) 25–34.
47. Skowron, A., J. Stepaniuk, S.: Tolerance Approximation Spaces. *Fundamenta Informaticae* **27** (1996) 245–253.
48. Skowron, A., Stepaniuk, J.: Information Granules and Rough-Neural Computing. In [29] (2004) 43–84.
49. Skowron, A., Stepaniuk, J.: Ontological framework for approximation. In [53] (2005) 718–727.
50. Skowron, A., Swiniarski, R.: Rough sets and higher order vagueness. In [53] (2005) 33–42.
51. Skowron, A., Swiniarski, R., Synak, P.: Approximation spaces and information granulation. In [35] (2004) 175–189.
52. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae **53** (2002) 365–387.
53. Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (Eds.): *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing* (RSFDGrC'2005), Regina, Canada, August 31-September 3, 2005, Part I, Lecture Notes in Artificial Intelligence, vol. 3641. Springer-Verlag, Heidelberg, Germany (2005).
54. Ślęzak, D., Yao, J.T., Peters, J.F., Ziarko, W., Hu, X. (Eds.): *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing* (RSFDGrC'2005), Regina, Canada, August 31-September 3, 2005, Part II, Lecture Notes in Artificial Intelligence, vol. 3642. Springer-Verlag, Heidelberg, Germany (2005).
55. Stepaniuk, J., Bazan, J., Skowron, A.: Modelling complex patterns by information systems. *Fundamenta Informaticae* **67**(1-3) (2005) 203–217.

56. Stepaniuk, J., Skowron, A., Peters, J.F., Swiniarski, R.: Calculi of approximation spaces. *Fundamenta Informaticae* **72**(1-3) (2206) (to appear).
57. Stone, P.: *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer.* The MIT Press, Cambridge, MA (2000).
58. Sun, R. (Ed.): *Cognition and Multi-Agent Interaction. From Cognitive Modeling to Social Simulation.* Cambridge University Press, New York, NY (2006).
59. Suraj, Z.: The synthesis problem of concurrent systems specified by dynamic information systems. In L. Polkowski, A. Skowron (Eds.), *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica-Verlag, Heidelberg (1998) 418–448.
60. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA (1998).
61. Utgoff, P.: An incremental ID3. In *Proceedings of the Fifth International Workshop on Machine Learning*, Ann Arbor, Michigan, June 12-14, 1988, Morgan Kaufmann (1988) 107–120.
62. Van Wezel, W., Jorna, R., Meystel, A.: *Planning in Intelligent Systems: Aspects, Motivations, and Methods.* John Wiley & Sons, Hoboken, New Jersey (2006).
63. Vapnik, V.: *Statistical Learning Theory.* John Wiley & Sons, New York, NY (1998).
64. Wang, G., Liu, Q., Yao, Y., Skowron, A. (Eds.): *Proceedings of the 9-th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing* (RSFDGrC'2003), Chongqing, China, May 26-29, 2003, Lecture Notes in Artificial Intelligence **2639**, Springer-Verlag, Heidelberg, Germany (2003).
65. Wojna, A.: Constraint Based Incremental Learning of Classification Rules. In [71] (2001) 428-435.
66. Zadeh, L.A.: Fuzzy sets. *Information and Control* **8** (1965) 333–353.
67. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. *Trans. on Systems, Man and Cybernetics* **SMC-3** (1973) 28-44.
68. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning I, II, III, *Information Sciences* **8** (1975) 199-257, 301-357; **9** 43-80.
69. Zadeh, L.A.: Fuzzy Logic = Computing with Words. *IEEE Transactions on Fuzzy Systems* **2** (1996) 103–111.
70. Ziarko, W.: Variable Precision Rough Set Model. *Journal of Computer and System Sciences* **46** (1993) 39–59.
71. Ziarko, W., Yao, Y. (Eds.): Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC'2000), Banff, Canada, October 16-19, 2000, Lecture Notes in Artificial Intelligence, vol. 2005. Springer-Verlag, Heidelberg, Germany, 2001.