Rough-Granular Computing in Human-Centric Information Processing

Tuan Trung Nguyen and Andrzej Skowron

Polish-Japanese Institute of Information Technology, Koszykowa 86,
02-008 Warsaw, Poland
nttrung@pjwstk.edu.pl
Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

Abstract. In ubiquitous computing, users are expected to continuously interact with computing devices, to suggest strategies and hypotheses, to pass over new facts from domain knowledge, to explain untypical cases in dialogs with the devices, etc. These devices therefore need to, at least in an approximate sense, understand the compound, vague concepts used by humans. We discuss current results and research directions on the approximation of compound vague concepts, which are based on rough-granular computing. In particular, we use hierarchical methods for the approximation of domain ontologies of vague concepts. We also discuss an extension of the proposed approach for approximate reasoning about interactive computations performed on complex granules by systems of agents in dynamically changing environments.

1 Selected Basic Issues on Granular Computing

In this section, we discuss some basic issue of *Granular Computing* (GC). We consider granules as constructive definitions of sets used in assembling objects satisfying a given specification at least to satisfactory degree. Granules are usually defined by granule systems [51, 52] in which some elementary granules are distinguished, together with operations making it possible to define new granules from these elementary granules, or from already defined granules. Among special types of operations on granules, one can distinguish the fusion and decomposition operations. For more readings on GC, the reader is referred to [2, 35, 38, 39, 42].

1.1 Synthesis of Complex Objects Satisfying Vague Specifications

One of the central issues related to granules is the definition of inclusion relations and closeness relations (measures). The concept of rough inclusion from rough mereology [45] can be used as a starting point in searching for constructive measures of inclusion or closeness of granules. Note that these measures should be defined for granules with different complexity structures.

In real-life applications, we often deal with problems where not only is the information about objects partial, but also the specification of problems is written in natural language. Hence, such specifications involve vague or/and imperfect

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 1–30.
springerlink.com

concepts. Problems we are trying to solve can be characterized as searching for complex objects satisfying a given specification to a satisfactory degree [45]. These complex objects should be synthesized from more elementary ones using available operations. Moreover, usually only partial information about these objects and concepts used in the specifications are available.

In the following section, we discuss searching for relevant granules as a kind of optimization problem in GC.

1.2 Optimization in Discovery of Compound Granules

This section is based on the approach discussed in [20, 30].

The problem considered in this section is the evaluation of perception as a means of optimizing various tasks. The solution to this problem hearkens back to early research on rough set theory and approximation. For example, in 1982, Ewa Orłowska observed that approximation spaces serve as a formal counterpart of perception.

In this chapter, the evaluation of perception is at the level of approximation spaces. The quality of an approximation space relative to a given approximated set of objects is a function of the description length of an approximation of the set of objects and the approximation quality of this set. In granular computing (GC), the focus is on discovering granules satisfying selected criteria. These criteria take inspiration from the minimal description length (MDL) principle proposed by Jorma Rissanen in 1983. In this section, the role of approximation spaces in modeling compound granules satisfying such criteria is discussed.

First, we recall the definition of an approximation space from [50]. Approximation spaces can be treated as granules used for concept approximation. They are examples of special parameterized relational structures. Tuning parameters make it possible to search for relevant approximation spaces relative to given concepts.

Definition 1. *A parameterized approximation space is a system* $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$, *where*

- U *is a non-empty set of objects,*
- $I_{\#} : U \to P(U)$ *is an uncertainty function, where* $P(U)$ *denotes the power set of* U,
- $\nu_{\$} : P(U) \times P(U) \to [0, 1]$ *is a rough inclusion function,*

and $\#, \$$ *denote vectors of parameters (the indexes* $\#, \$$ *will be omitted if it does not lead to misunderstanding).*

The uncertainty function defines for every object x, a set of objects described similarly to x. The set $I(x)$ is called the neighborhood of x (see, *e.g.*, [36, 50]).

The rough inclusion function $\nu_{\$} : P(U) \times P(U) \to [0, 1]$ defines the degree of inclusion of X in Y, where $X, Y \subseteq U$.

In the simplest case it can be defined by (see, *e.g.*, [50, 36]):

$$\nu_{SRI}(X, Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)}, & \text{if } X \neq \emptyset, \\ 1, & \text{if } X = \emptyset. \end{cases}$$

The lower and the upper approximations of subsets of U are defined as follows.

Definition 2. *For any approximation space $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ and any subset $X \subseteq U$, the lower and upper approximations are defined by*
$LOW\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}\left(x\right), X\right) = 1\}$,
$UPP\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}\left(x\right), X\right) > 0\}$, *respectively.*

The lower approximation of a set X with respect to the approximation space $AS_{\#,\$}$ is the set of all objects that can be classified with certainty as objects of X with respect to $AS_{\#,\$}$. The upper approximation of a set X with respect to the approximation space $AS_{\#,\$}$ is the set of all objects which can be possibly classified as objects of X with respect to $AS_{\#,\$}$.

Several known approaches to concept approximation can be covered using this approach to approximation spaces (see, *e.g.*, references in [50]). For more details on approximation spaces, the reader is referred to, *e.g.*, [10, 37, 53, 41, 44].

A key task in granular computing is the information granulation process that leads to the formation of information aggregates (with inherent patterns) from a set of available objects. A methodological and algorithmic issue is the formation of transparent (understandable) information granules inasmuch as they should provide a clear and understandable description of patterns present in sample objects [2, 39]. Such a fundamental property can be formalized by a set of constraints that must be satisfied during the information granulation process. Usefulness of these constraints is measured by the quality of an approximation space:

$$Quality_1 : Set\_AS \times P(U) \rightarrow [0, 1],$$

where U is a non-empty set of objects and Set_AS is a set of possible approximation spaces with the universe U.

Example 1. If $UPP(AS, X)) \neq \emptyset$ for $AS \in Set\_AS$ and $X \subseteq U$ then

$$Quality_1(AS, X) = \nu_{SRI}(UPP(AS, X), LOW(AS, X)) = \frac{card(LOW(AS, X))}{card(UPP(AS, X))}.$$

The value $1 - Quality_1(AS, X)$ expresses the degree of completeness of our knowledge about X, given the approximation space AS.

Example 2. In applications, we usually use another quality measure analogous to the minimal length principle [47, 56] where also the description length of approximation is included. Let us denote by $description(AS, X)$ the description length of approximation of X in AS. The description length may be measured, *e.g.*, by the sum of description lengths of algorithms testing membership for neighborhoods used in construction of the lower approximation, the upper approximation, and the boundary region of the set X. Then the quality $Quality_2(AS, X)$ can be defined by

$$Quality_2(AS, X) = g(Quality_1(AS, X), description(AS, X)),$$

where g is a relevant function used for fusion of values $Quality_1(AS, X)$ and $description(AS, X)$. This function g, for instance, may involve weights assigned by experts to both criteria.

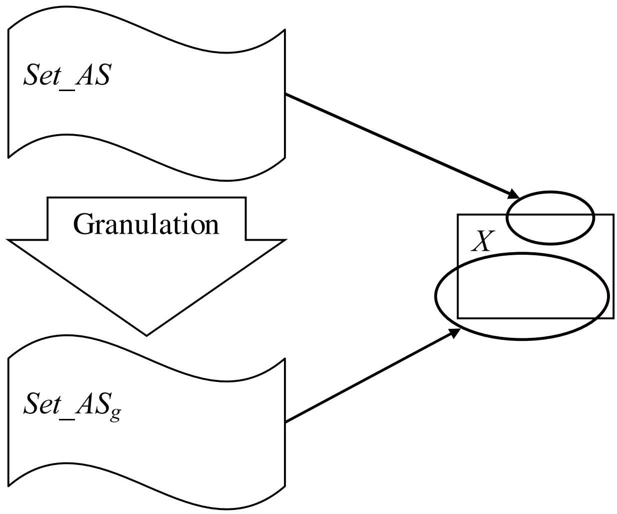

Fig. 1. Granulation of parameterized approximation spaces

One can consider different optimization problems relative to a given class Set_AS of approximation spaces. For example, for a given $X \subseteq U$ and a threshold $t \in [0, 1]$, one can search for an approximation space AS satisfying the constraint $Quality_2(AS, X) \geq t$.

Another example involves searching for an approximation space satisfying additionally the constraint $Cost(AS) < c$ where $Cost(AS)$ denotes the cost of an approximation space AS (*e.g.*, measured by the number of attributes used to define neighborhoods in AS) and c is a given threshold. In the following example, we consider also costs of searching for relevant approximation spaces in a given family defined by a parameterized approximation space (see Figure 1). Any parameterized approximation space $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ is a family of approximation spaces. The cost of searching in such a family for a relevant approximation space for a given concept X approximation can be treated as a factor of the quality measure of approximation of X in $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$. Hence, such a quality measure of approximation of X in $AS_{\#,\$}$ can be defined by

$$Quality_3(AS_{\#,\$}, X) = h(Quality_2(AS, X), Cost\_Search(AS_{\#,\$}, X)),$$

where AS is the result of searching in $AS_{\#,\$}$, $Cost\_Search(AS_{\#,\$}, X)$ is the cost of searching in $AS_{\#,\$}$ for AS, and h is a fusion function, *e.g.*, assuming that the values of $Quality_2(AS, X)$ and $Cost\_Search(AS_{\#,\$}, X)$ are normalized to interval $[0, 1]$ h could be defined by a linear combination of $Quality_2(AS, X)$ and $Cost\_Search(AS_{\#,\$}, X)$ of the form

$$\lambda Quality_2(AS, X) + (1 - \lambda) Cost\_Search(AS_{\#,\$}, X),$$

where $0 \leq \lambda \leq 1$ is a weight measuring an importance of quality and cost in their fusion.

We assume that the fusion functions g, h in the definitions of quality are monotonic relative to each argument.

Let $AS \in Set\_AS$ be an approximation space relevant for approximation of $X \subseteq U$, *i.e.*, AS is the optimal (or semi-optimal) relative to $Quality_2$. By $Granulation(AS_{\#,\$})$ we denote a new parameterized approximation space obtained by granulation of $AS_{\#,\$}$. For example, $Granulation(AS_{\#,\$})$ can be obtained by reducing the number of attributes or inclusion degrees (*i.e.*, possible values of the inclusion function). Let AS' be an approximation space in $Granulation(AS_{\#,\$})$ obtained as the result of searching for optimal (semi-optimal) approximation space in $Granulation(AS_{\#,\$})$ for approximation of X.

We assume that three conditions are satisfied:

- after granulation of $AS_{\#,\$}$ to $Granulation(AS_{\#,\$})$ the following property holds: the cost

$$Cost\_Search(Granulation(AS_{\#,\$}), X),$$

  is much lower than the cost $Cost\_Search(AS_{\#,\$}, X)$;
- The $description(AS', X)$ is much shorter than the $description(AS, X)$, *i.e.*, the description length of X in the approximation space AS' is much shorter than the description length of X in the approximation space AS;
- $Quality_1(AS, X)$ and $Quality_1(AS', X)$ are sufficiently close.

The last two conditions should guarantee that the values $Quality_2(AS, X)$ and $Quality_2(AS', X)$ are comparable and this condition together with the first condition about the cost of searching should assure that

$$Quality_3(Granulation(AS_{\#,\$}, X)) \text{ is much better than } Quality_3(AS_{\#,\$}, X).$$

Taking into account that parameterized approximation spaces are examples of parameterized granules, one can generalize the above example of parameterized approximation space granulation to the case of granulation of parameterized granules.

In the process of searching for (sub-)optimal approximation spaces, different strategies may be used. Let us consider an example of such strategies [55]. In the example, $DT = (U, A, d)$ denotes a decision system (a given sample of data), where U is a set of objects, A is a set of attributes and d is a decision. We assume that for any object x, only partial information, equal to the A-signature of x (object signature, for short), is available, *i.e.*, $Inf_A(x) = \{(a, a(x)) : a \in A\}$ and analogously for any concept, only partial information about this concept by a sample of objects is provided, *e.g.*, in the form of decision table. One can use object signatures as new objects in a new relational structure $\mathcal{R}$. In this relational structure $\mathcal{R}$ some relations between object signatures are also modelled, *e.g.*, defined by the similarities of these object signatures. Discovery of relevant relations on object signatures is an important step in searching for relevant approximation spaces. In this way, a class of relational structures representing perception of objects and their parts is constructed. In the next step, we select a language $\mathcal{L}$ of formulas expressing properties over the defined relational structures and we search for relevant formulas in $\mathcal{L}$. The semantics of formulas (*e.g.*, with one free variable) from $\mathcal{L}$ are subsets of object signatures. Observe that each object signature defines a neighborhood of objects from a given sample (*e.g.*, decision table DT) and another set on the whole universe of objects being an extension of U. Thus, each formula from $\mathcal{L}$ defines a family of sets of objects over the sample and also another family of sets over the universe of all objects. One can use such families can to define new neighborhoods of a new approximation space, *e.g.*, by taking their unions. In the searching process for relevant neighborhoods, we use information encoded in the given sample. More relevant neighborhoods make it possible to define relevant approximation spaces (from the point of view of the optimization criterion). It is worth to mention that often this searching process is even more sophisticated. For example, one can discover several relational structures (e.g., corresponding to different attributes) and formulas over such structures defining different families of neighborhoods from the original approximation space. Next such families of neighborhoods can be merged into neighborhoods in a new approximation space. This kind of modeling is typical for hierarchical modeling [8], *e.g.*, when we search for a relevant approximation space for objects composed from parts for which some relevant approximation spaces have been already found.

2 Granular Computing and Human Perception: Learning in Dialog with Human Experts

The hierarchical learning approach takes advantage of additional domain knowledge provided by human experts. In order to best employ this knowledge, it relies on the observation that human thinking and perception in general, and their reasoning while performing classification tasks in particular, can:

- inherently comprise different levels of abstraction,
- display a natural ability to switch focus from one level to another,
- operate on several levels simultaneously.

Such processes are natural subjects for the *Granular Computing* paradigm, which encompasses theories, methods, techniques and tools for such fields as problem solving, information processing, human perception evaluation, analysis of complex systems and many others. It is built around the concept of *information granules*, which can be understood as collections of *values that are drawn together by indistinguishability, equivalence, similarity, or proximity* [63]. Granular Computing follows the human ability to perceive things in different levels of abstraction (*granularity*), to concentrate on a particular level of interest while preserving the ability to instantly switch to another level in case of need. This allows to obtain different levels of knowledge and, which is important, a better understanding of the inherent structure of this knowledge.

The concept of information granules is closely related to the imprecise nature of human reasoning and perception. Granular Computing therefore provides excellent tools and methodologies for problems involving flexible operations on imprecise or approximated concepts expressed in natural language.

One of the possible approaches in developing methods for compound concept approximations can be based on the layered (hierarchical) learning [11, 57]. Inducing concept approximation should be developed hierarchically starting from concepts that can be directly approximated using sensor measurements toward compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use some rules of behavior on the roads, expressed in natural language, to assess from recordings (made, e.g., by camera and other sensors) of actual traffic situations, if a particular situation is safe or not (see, e.g., [8, 9, 14, 31]). The hierarchical learning has been also used for identification of risk patterns in medical data and extended for therapy planning (see, e.g. [6, 7]). Another application of hierarchical learning for sunspot classification is reported in [33]. To deal with such problems one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. The foundations of such an approach, creating a core of perception logic, are based on rough set theory [14, 36, 37] and its extension rough mereology [35, 45, 51]. The (approximate) Boolean reasoning methods can be scaled to the case of compound concept approximation.

Let us observe that hierarchical modeling employs some general mechanisms emphasized in [22] dealing with a kind of “interplay” between syntax and semantics. The key observation is that the syntax on one level is used to define semantical structures (or their clusters) on the next level of hierarchy. One can interpret them in the framework of the Bairwise classifications [4] as operations on such classifications or as a kind of sums of information systems [54]. They allow us gradually to model structures of granules representing “wider” context of perceived objects. In this way, it is possible to construct more compound granules interpreted, e.g., as patterns representing properties of, e.g., time windows of states, sequences of such time windows, sets of such sequences, etc.

2.1 Hierarchical Modeling and Dealing with Ill-Posed Problems: Toward Generalization of the Minimal Length Principle to the Case of Concept Ontology

As pointed out in [61], machine learning problems can be considered as inverse problems, and in a broad view,

$$A(f) = d,$$

where A can be understood as a model for a phenomena, $f \in \mathcal{F}$ represents a function of some of the model's causal factors, chosen from a class $\mathcal{F}$ of candidate functions, and d denotes some actual observation data pertaining to the phenomena, are generally ill-posed, which means the solution f might not exist, might not be unique, and most importantly, might not be stable. Namely, with a small deviation δ in the output data d_δ, we have

$$R_\delta(f) = \|A(f) - d_\delta\|, \tag{1}$$

not tending to zero even if δ tends to zero, where $\| \cdot \|$ is any divergence metrics appropriate for f, meaning arbitrarily small deviations in data may cause large deviations in solutions.

One can also give another interpretation of the equation (1). The operator A can be interpreted as a (vague) specification (constraints) of the problem and the goal is to find a solution f satisfying the specification to a satisfactory degree. This satisfactory degree is expressed in (1) by means of the norm. Note that, very often, while dealing with real-life problems we have only a vague specification A rather than a crisp operator A. Moreover, due to the uncertainty in specification of A and f the quality measures often can only be estimated from available data. In consequence, one can hardly expect that the relevant measures would be expressed in well known spaces with norms as in (1). In such cases one should look for some other avenues to express, e.g., the phrase *a solution should satisfy a given specification to satisfactory degree* [30].

For dealing with ill posed problems the regularization theory was proposed. The idea of regularization is due to Tikhonov (1963, see [60]). Instead of the equation (1) the following one is considered:

$$R_{\delta,\gamma}(f) = \|A(f) - d_\delta\| + \gamma W(f), \tag{2}$$

where $W(f)$ is a functional measuring the “simplicity” of the solution f and γ is a parameter (adjustable in the learning process).

Now, in the equation (2) we have a sum of two arguments. The first one expresses the quality of the solution f and the second one expresses, in a sense, the description length of the solution, using the terminology related to the minimal length principle. For a given parameter γ we are searching for f by minimizing the value of $R_{\delta,\gamma}(f)$. By choosing different values of γ we may alter our priority given to the first or the second summand of the sum in (2).

Fundamental pattern recognitions problems such as class probability density function estimation from a wide set of potential densities, or parametric estimation of optimal feature subsets, are ill-posed.

On the other hand, if the model A can be decomposed into a combination of simpler sub-models A_i, e.g. those involving search spaces with lower Vapnik-Chervonenkis (VC) dimensions, or those for which respective stable sub-solutions f_i can be found inexpensively, chances are that we'll be able to assemble a solution f from sub-solutions f_i, which will be better than a solution computed in an all-out attempt for the original problem. However, the challenge in this approach is that there is no known automatic method for the computation of effective decompositions of A.

In the hierarchical learning approach, we assume that the decomposition scheme will be provided by an external human expert in an interactive process. Knowledge acquired from human expert will serve as guidance to break the original model A into simpler, more manageable sub-models A_i, organized in a lattice-like hierarchy. They would correspond to subsequent levels of abstractions in the hierarchy of perception and reasoning of the human expert.

The mentioned above decomposition should lead to submodels A_i together with pertaining functionals W_i as well as parameters γ_i. The global optimization criteria become more compound in the decomposition case and should be obtained by fusion of those for submodels. For example, one could assume the following optimization criterion:

$$R^*_{\delta,\gamma}(f) = \sum_i \|A_i(f_i) - d_{\delta_i}\| + \gamma_i W_i(f_i), \tag{3}$$

where the sum is taken over all decomposition submodels and f is the solution corresponding to the root level of decomposition (i.e., to the model A)[1]. However, the linear fusion in (3) may be too simplistic for real-life problems, where it is important to learn from data approximations of optimization criteria [25, 30].

2.2 Narrowing the Potential Search Space

As stated in [61], the problem of estimating f from a large set $\mathcal{F}$ of possible candidate solutions is ill-posed. One way to alleviate this problem is to employ the so-called Structural Risk Minimization (SRM) technique. The technique, in short, is based on a theorem on the risk's bounds, which essentially states that

$$R(\alpha) \leq R_{emp}(\alpha) + CI(\alpha),$$

which means the risk functional $R(\alpha)$, expressing how far we are from the desired solution for a parameter α from a general parameter set S, is bounded by the sum of the empirical risk $R_{emp}(\alpha)$ and a confidence interval $CI(\alpha)$ containing the Vapnik-Chervonenkiss dimension of the function space S.

This dependency is shown on Fig. 2.

Instead of optimizing α over an arbitrary set of possible parameters S, we use the bounds to find a set S^* for which the risk's bound is minimal, and then perform the search for the solution α^* within S^*. For more details, see [61].

The hierarchical learning approach, by reducing the complexity of the original learning problem by decomposing it into simpler ones, tries to optimize the corresponding search spaces on subsequent levels of the learning hierarchy, and is analogous in function to the SRM technique. One can consider decomposition as one of possible strategies in SRM aimed at searching for (sub)optimal spaces. The resulting space corresponds to the family of searching spaces obtained on different levels of decomposition. For any submodel on the $i+1$-th level the searching space for solutions is discovered on the basis of some search spaces from the i-th level. The search for (sub)optimal decomposition is conducted by minimization of the description length of solutions from spaces on different decomposition levels while preserving the satisfactory quality of solutions. The searching spaces for approximation of concepts from any level $i+1$ on the basis of concepts from the level i of decomposition are computationally feasible because any two successive levels of decomposition should be, in a sense, semantically *close* [30]. This means that the searching spaces brought about on particular

[1] Searching for any f_i (not corresponding to the leaf decomposition level) is performed over the space constructed on the basis of some already discovered spaces linked to some submodels from the predecessor decomposition level relative to f_i.

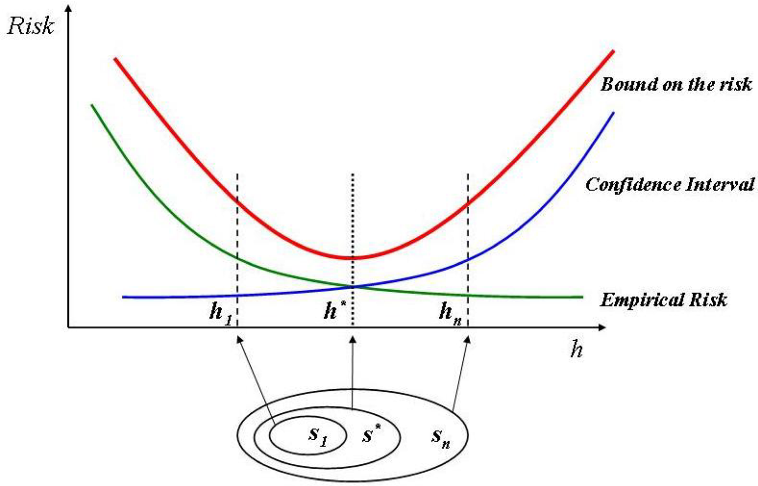

Fig. 2. Actual risk bounds across search spaces.(Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1999)

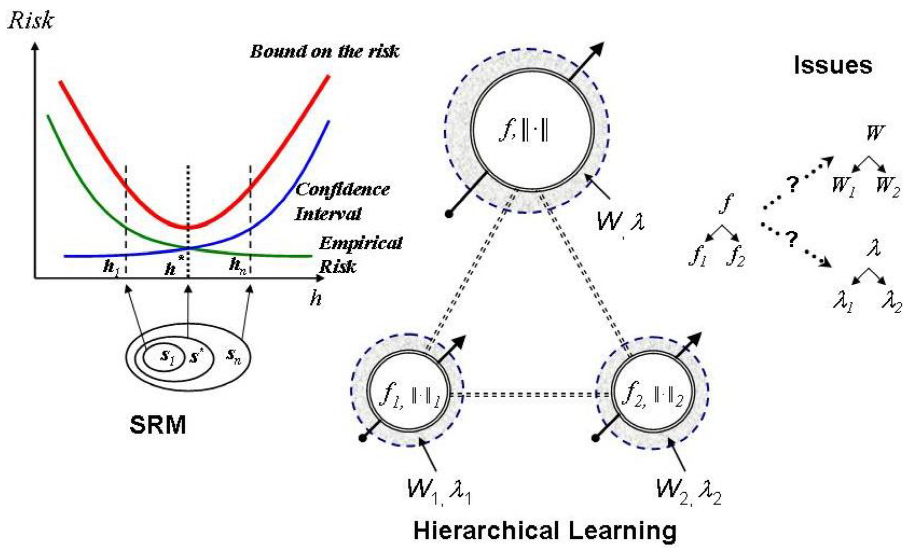

Fig. 3. SRM vs Hierarchical Learning

decomposition levels are smaller than those obtained without decomposition. Moreover SRM can be used on each particular decomposition level to optimize the searching space for approximation of concepts on this level. For details, see Fig. 3.

Another advantage of the hierarchical learning model lies in the construction of the descriptive language in which classifiers on subsequent levels are built. The choice of language directly influences the potential search space and is therefore crucial for classifier building. With a proper reasoning scheme in place, we can construct the descriptive language on a higher level from those already established on lower levels, which proves effective in reducing the learning time and boosting the overall learning performance. The choice of language can be interpreted as a step in searching for sub(optimal) spaces in SRM.

2.3 Ontology Matching

The knowledge on training samples that comes from an expert obviously reflects his perception about the samples. The language used to describe this knowledge is a component of the expert's ontology which is an integral part of his perception. In a broad view, an ontology consists of a vocabulary, a set of concepts organized in some kind of structures, and a set of binding relations amongst those concepts [15]. We assume that the expert's ontology when reasoning about complex structured samples will have the form of a multi-layered hierarchy, or a *lattice*, of concepts. A concept on a higher level will be synthesized from its children concepts and their binding relations. The reasoning thus proceeds from the most primitive notions at the lowest levels and work bottom-up towards more complex concepts at higher levels.

Hierarchical learning, together with the transfer of knowledge expressed in natural languages from external experts to low-level computer operators, constitutes an excellent illustration of *Granular Computing* in action.

2.4 External Knowledge Transfer

The knowledge elicitation process assumes that samples, for which the learning system deems it needs additional explanations, are submitted to the expert, which returns not only their correct class identity, but also an explanation on *why*, and perhaps more importantly, *how* he arrived at his decision. This explanation is passed in the form of a rule:

$$[CLASS(u) = k] \equiv \Im(EFeature_1(u), ..., EFeature_n(u)),$$

where $EFeature_i$ represents the expert's perception of some characteristics of the sample u, while synthesis operator $\Im$ represents his perception of some relations between these characteristics. In a broader view, $\Im$ constitutes of a *relational structure* that encompasses the hierarchy of experts' concepts expressed by $EFeature_i$.

The ontology matching aims to translate the components of the expert's ontology, such as $EFeature_i$ and binding relations embedded in the $\Im$ structure, expressed in the foreign language L_f, into the patterns (or classifiers) expressed in a language familiar to the learning system, e.g:

- $[FaceType(Ed) = Square] \equiv$(Ed.Face().Width - Ed.Face().Height) $\leq$ 2cm,
- $[Eclipse(p) = True] \equiv$ (s=p.Sun())$\wedge$(m=p.Moon())$\wedge$(s$\cap$m.Area$\geq$ s.Area$\cdot$0.6).

Here the abstract concepts such as "Ed has a square face" or "The Sun is in eclipse" get translated into classification rules built from computable measurements and observation features.

As the human perception is inherently prone to variation and deviation, the concepts and relations in a human expert's ontology are approximate by design. To use the terms of granular computing, they are information granules that encapsulate the autonomous yet interdependent aspects of human perception.

The matching process, while seeking to accommodate various degrees of variation and tolerance in approximating those concepts and relations, will follow the same hierarchical structure of the expert's reasoning. This allows parent concepts to be approximated using the approximations of children concepts, essentially building a *layered approximate reasoning scheme*. Its hierarchical structure provides a natural realization of the concept of granularity, where nodes represent clusters of samples/classifiers that are similar within a degree of resemblance/functionality, while layers form different levels of abstraction/perspectives on selected aspects of the sample domain.

On the other hand, with such an established multi-layered reasoning architecture, we can take advantages of the results obtained within the Granular Computing paradigm, which provides frameworks and tools for the fusion and analysis of compound information granules from previously established ones, in a straightforward manner. The intermediate concepts used by external experts to explain their perception are vague and ambiguous, which makes them natural subjects to granular calculi.

The translation must

- allow for a flexible matching of a variations of similar domestic patterns to a foreign concept, i.e. the translation result should not be a single patterns, but rather a collection or cluster of patterns.
- find approximations for the foreign concepts and relations, while preserving their hierarchical structure. In other words, inherent structure of the provided knowledge should be intact.
- ensure robustness, which means independence from noisy input data and incidental underperformance of approximation on lower levels, and stability, which guarantees that any input pattern matching concepts on a lower level to a satisfactory degree will result in a satisfactory target pattern on the next level.

We assume an architecture that allows a learning system to consult a human expert for advices on how to analyze a particular sample or a set of samples. Typically this is done in an iterative process, with the system subsequently incorporating knowledge elicited on samples that could not be properly classified in previous attempts [32]. (See Fig. 4 below).

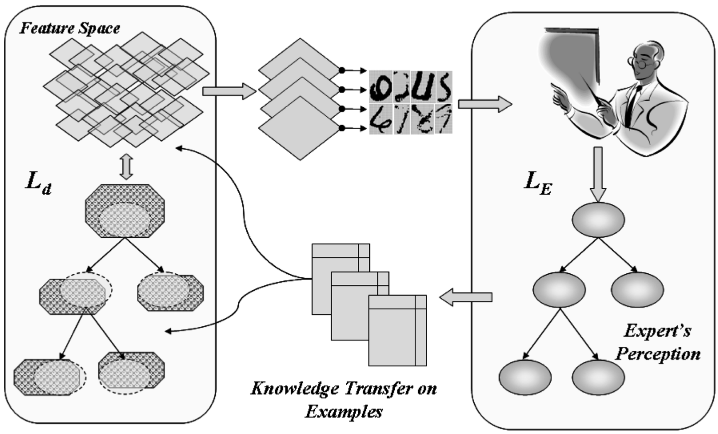

Fig. 4. Expert's knowledge elicitation

2.4.1 Approximation of Concepts

A foreign concept C is approximated by a domestic pattern (or a set of patterns) p in term of a rough inclusion measure $Match(p, C) \in [0, 1]$. Such measures take root in the theory of rough mereology [45], and are designed to deal with the notion of inclusion to a degree. An example of concept inclusion measures would be:

$$Match(p, C) = \frac{|\{u \in T : Found(p, u) \wedge Fit(C, u)\}|}{|\{u \in T : Fit(C, u)\}|},$$

where T is a common set of samples used by both the system and the expert to communicate with each other on the nature of expert's concepts, $Found(p, u)$ means a pattern p is present in u and $Fit(C, u)$ means u is regarded by the expert as fit to his concept C.

Our principal goal is, for each expert's explanation, find sets of patterns Pat, Pat_1,...,Pat_n and a relation $\Im_d$ so as to satisfy the following *quality requirement*:

$$\textbf{if } (\forall i : Match(Pat_i, EFeature_i) \geq p_i) \wedge (Pat = \Im_d(Pat_1, ..., Pat_n))$$
$$\textbf{then } Quality(Pat) > \alpha,$$

where $p, p_i : i \in \{1, .., n\}$ and α are certain cutoff thresholds, while the $Quality$ measure, intended to verify if the target pattern Pat fits into the expert's concept of sample class k, can be any, or combination, of popular quality criteria such as *support*, *coverage*, or *confidence* [46], where

$$Support_{CLASS=k}(Pat) = |\{u \in U : Found(Pat, u) \wedge CLASS(u) = k\}|,$$

$$Confidence_{CLASS=k}(Pat) = \frac{Support(Pat)}{|\{u \in U : Found(Pat, u)\}|},$$

$$Coverage_{CLASS=k}(Pat) = \frac{Support(Pat)}{|\{u \in U : CLASS(u) = k\}|},$$

and U is the training set.

In other words, we seek to translate the expert's knowledge into the domestic language so that to generalize the expert's reasoning to the largest possible number of training samples. More refined versions of the inclusion measures would involve additional coefficients attached to e.g. $Found$ and Fit test function. Adjustment of these coefficients based on feedback from actual data may help optimize the approximation quality.

For example, let's consider a handwritten digit recognition task:

When explaining his perception of a particular digit image sample, the expert may employ concepts such as *Circle*, *Vertical Strokes* or *West Open Belly*. The expert will explain what he means when he says, e.g. $Circle$, by providing a decision table (U, d) with reference samples, where d is the expert decision to which degree he considers that $Circle$ appears in samples $u \in U$. The samples in U may be provided by the expert, or may be picked up by him among samples explicitly submitted by the system, e.g. those that had been misclassified in previous attempts.

The use of rough inclusion measures allows for a very flexible approximation of foreign concept. A stroke at 85 degree to the horizontal in a sample image can still be regarded as a vertical stroke, though obviously not a 'pure' one. Instead of just answering in a Yes/No fashion, the expert may express his degrees of belief using such natural language terms as $Strong$, $Fair$, or $Weak$ (See Fig. 5).

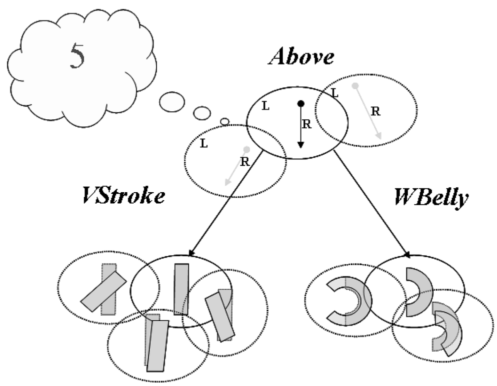

Fig. 5. Tolerant matching by expert

The expert's feedback will come in the form of a decision table (See Table 1).

Table 1. Perceived features

	$Circle$
u_1	$Strong$
u_2	$Weak$
...	...
u_n	$Fair$

Table 2. Translated features

	$DPat$	$Circle$
u_1	252	$Strong$
u_2	4	$Weak$
...	...	...
u_n	90	$Fair$

The translation process attempts to find domestic feature(s)/pattern(s) that approximate these degrees of belief (e.g. such as presented in Table 2). Domestic patterns satisfying the defined quality requirement can be quickly found, taking into account that sample tables submitted to experts are usually not very large. Since this is essentially a rather simple learning task that involves feature selection, many strategies can be employed. In [34], genetic algorithms equipped with some greedy heuristics are reported successful for a similar problem. Neural networks also prove suitable for effective implementation.

It can be observed that the intermediate concepts like *Circle* or *Vertical Strokes*, provided by a human expert, along with satisfiability assessments like $Strong$, $Fair$, or $Weak$ form information granules within the perception of the expert. The granules correspond to different levels of abstraction, or focus, of his reasoning about a particular class of samples. The translation process transforms these information granules into classifiers capable of matching particular parts of actual samples with intermediate expert's concepts, which essentially incorporates the human perception, by way of using information granules, into the learning process.

2.4.2 Approximation of Relations

The approximation of higher level relations between concepts has been formalized within the framework of perception structures, recently developed in [49]. A *perception structure* S, in a simpler form, is defined as:

$$S = (U, M, F, \models, p),$$

where U is a set of samples, F is a family of formulas expressed in domestic language that describe certain features of the samples and M is a family of relational structures in which these formulas can be evaluated, while $p\colon U \to M \times F$ is a *perception function* such that $\forall u \in U : p_1(u) \models p_2(u)$ (p_1 and p_2 are the first and second component projections of p) which means that $p_2(u)$ is satisfied (is true) in the relational structure $p_1(u)$. This may express that some relations among features within samples are observed.

For a given sample u, we define a set

$$M(u) = \{R \in M : R \models p_2(u)\},$$

which contains all possible relational structures for which formulas, or in other words, features observed in u yield.

2.4.3 Approximate Clusters

Given a perception structure S, an approximate cluster of a given sample u is defined as:

$$[u]_S = \bigcup_{R \in M(u)} p_1^{-1}(R).$$

This cluster contains samples from U that have similar structures to u, with regard to the perception p, i.e. those with similar relational structures that also hold true the features observed in u (See Fig. 6).

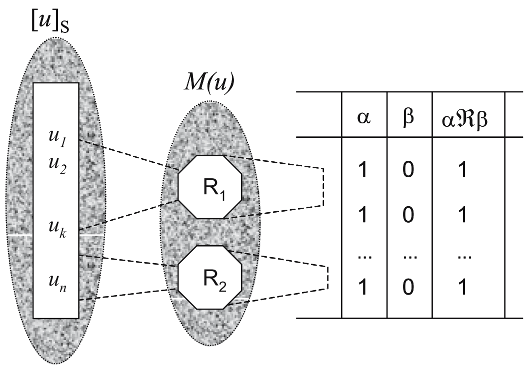

Fig. 6. Approximate cluster

For example, if we construct a perception structure that contains a formula describing a part of a digit is *above* another part, then within this perception, the approximate cluster of a digit '6', which has a slant stroke over a circle, would comprise of all digits that have similar structure, i.e. containing a slant stroke over a circle.

Perception structures, following natural constructs in the expert's foreign language, should involve tolerant matching. Let's suppose that we allow a *soft* perception on samples of U by introducing a similarity relation τ between them. This relation, for example might assume that two samples resemble each other to a degree. This naturally leads to clusters of similar relational structures in M. With samples now perceived as similar to each other in a degree, we shall allow for a similarity relation in M. Two relational structures might be considered approximately *the same* if they allow for similar formulas to yield similar results in majority of cases when these formulas are applicable. The family M thus becomes granulated by τ and is denoted by M_τ.

The same follows for the family F of features, or formulas that, for instance, do not always have the same value, but are equivalent in most cases, or in all or majority of a cluster of similar relational structures. Formulas' evaluation might be extended to comprise degrees of truth values, rather than plain binary constants. The family F hence becomes granulated with regards to τ and is denoted by F_τ.

The perception structure S hence becomes, for a given similarity measure τ in U: $S = (U, M_\tau, F_\tau, \models, p)$ which permits a much more flexible space and a variety of methods for concept approximation.

In the above mentioned example, a similarity induced perception might consider as the approximate cluster of a digit '5' the set of every samples that will have a *stroke* over a *closed curve* (not just slant strokes and circles as before). Moreover, the new perception also allows for a greater variation of configurations considered to fit into the concept of *above*.

The definition of an approximate cluster becomes:

$$[u]_S = \bigcup_{R \in M_\tau(u)} p_1^{-1}(R).$$

The task of approximating an expert's concept involving relations between components is now equivalent to finding a perception function that satisfies some quality criteria. Let's suppose that the expert provide us a set C of samples he considers fit to his concept. We have to find a perception function p such that:

$$Confidence : \frac{|[u]_S \cap C|}{|[u]_S|} > c,$$

and/or

$$Support : \frac{|[u]_S \cap C|}{|U|} > s,$$

where u is some sample from C, and $0 < c, s < 1$.

Having approximated the expert's features $EFeature_i$, we can try to translate his relation $\Im$ into our $\Im_d$ by asking the expert to go through U and provide us with the additional attributes of how strongly he considers the presence of $EFeature_i$ and to what degree he believes the relation $\Im$ holds. Again, lets consider the handwritten recognition case.(See Table 3).

Table 3. Perceived relations

	VStroke	*WBelly*	*Above*
u_1	*Strong*	*Strong*	*Strong*
u_2	*Fair*	*Weak*	*Weak*
...	...	...	...
u_n	*Fair*	*Fair*	*Weak*

Table 4. Translated relations

	#V_S	#NES	$S_y < B_y$	*Above*
u_1	0.8	0.9	(*Strong*,1.0)	(*Strong*, 0.9)
u_2	0.9	1.0	(*Weak*, 0.1)	(*Weak*, 0.1)
...	...	...	...	...
u_n	0.9	0.6	(*Fair*, 0.3)	(*Weak*, 0.2)

We then replace the attributes corresponding to $EFeature_i$ with the rough inclusion measures of the domestic feature sets that approximate those concepts (computed in the previous step). In the next stage, we try to add other features, possibly induced from original domestic primitives, in order to approximate the decision d. Such a feature may be expressed by $S_y < B_y$, which tells whether the median center of the stroke is placed closer to the upper edge of the image than the median center of the belly. (See Table 4).

The expert's perception *A '6' is something that has a 'vertical stroke' 'above' a 'belly open to the west'* is eventually approximated by a classifier in the form of a rule:

if S(#BL_SL > 23) **AND** B(#NESW > 12%) **AND** $S_y < B_y$ **then** CL='6',

where S and B are designations of pixel collections, #BL_SL and #NESW are numbers of pixels with particular topological feature codes, and $S_y < B_y$ reasons about centers of gravity of the two collections.

Approximate reasoning schemes embody the concept of information granularity by introducing a hierarchical structure of abstraction levels for the external knowledge that come in the form of a human expert's perception. The granularity helps to reduce the cost of the knowledge transfer process, taking advantage of the expert's hints. At the same time, the hierarchical structure ensures to preserve approximation quality criteria that would be hard to obtain in a flat, single-level learning process.

From yet another perspective, the reasoning schemes that encompass a human expert's intermediate concepts like *Vertical Stroke*, *Above* and their satisfability assessments such as *Strong* or *Fair* represents the way humans reason about samples through different levels of abstraction. The connections between intermediate concepts and transitions from lower to upper levels allow to shift the perception focus from smaller parts of objects to more abstract, global features. These reasoning schemes also provide off-the-shelf recipes as to how to assemble more compound information granules from simpler, already established ones. Translated into domestic languages, they become powerful classifiers that help expand the human perception structures to actual samples.

2.5 Outliers

Conceptually, *outliers/exceptions* are kind of atypical samples that stand out from the rest of their group or behave very differently from the norm [1]. While there is still no universally accepted formal definition of being an outlier, several descriptions seem to reflect the essential spirit. According to Hawkin: *An outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism*, while Barnett and Lewis define an outlier as *an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.* [3]. These samples previously would usually be treated as bias or noisy input data and were frequently discarded or suppressed in subsequent analyses. However, the rapid development of Data Mining, which aims to extract from data as much knowledge as possible, has made outlier identification and analysis one of its principal branches. Dealing with outliers is crucial to many important fields in real life such as fraud detection in electronic commerce, intrusion detection, network management, or even space exploration. At the same time, there is an increasing effort in the Machine Learning community to develop better methods for outlier detection/analysis, as outliers often carry useful subtle hints on the characteristics of the sample domain and, if properly analyzed, may provide valuable guidance in discovering the causalities underlying the behavior of a learning system. As such, they may prove valuable as an additional source of search control knowledge and as a mean for the construction of better classifiers.

Most popular measures to detect outliers [19] are based on either probabilistic density analysis [12] or distance evaluation [28]. Knorr made an attempt to elicit intensional knowledge from outliers through the analysis of the dynamicity of outliers' set against changes in attribute subsets [27]. However, no thorough model or scheme for the discovery of intensional knowledge from identified outliers has been established. In particular, there is almost no known attempt to develop methods for outlier analysis amongst structured objects, i.e. objects that display strong inner dependencies between theirs own features or components. Perhaps the reason for this is the fact that while many elaborated computation models for the detection of outliers have been proposed , their effective use in eliciting additional domain knowledge, as well as the elicitation of intensional knowledge within outliers, is believed difficult without support of a human expert.

In this paper, we approach the detection and analysis of outliers in data from a Machine Learning perspective. We propose a framework based on the Granular Computing paradigm, using tools and methods originated from Rough Set and Rough Mereology theories. The process of outlier detection is refined by the evaluation of classifiers constructed employing intensional knowledge elicited from suspicious samples. The internal structures of the sample domain will be dealt with using hierarchical approximate reasoning schemes and layered learning. We show the role of an external domain knowledge source by human experts in outlier analysis, and present methods for the successful assimilation of such knowledge. Introduced methods and schemes are illustrated with an example handwritten digit recognition system.

Most existing outlier identification methods employ either probabilistic density analysis, or distance measures evaluation [19]. Probabilistic approach typically run a series of statistical discordancy tests on a sample to determine whether it can be qualified as an outlier. Sometimes this procedure is enhanced by a dynamic learning process. Their main weakness is the assumption of an underlying distribution of samples, which is not always available in many real life applications. Difficulties with their scalability in numbers of samples and dimensions are also a setback of primary concern.

Another approach to outlier detection relies on certain distance measures established between samples. Known methods are data clustering and neighbor analysis. While this approach can be applied to data without any assumed a priori distribution, they usually entails significant computation costs.

Let C_k be a cluster of samples for class k during the training phase and d_k be the distance function established for that class. For a given cut-off coefficient $\alpha \in (0, 1]$, a sample u^* of class k is considered "difficult", "hard" or "outlier" if, e.g:

$$d_k(u^*, C_k) \geq \alpha \cdot \max\{(v, C_K) : v \in TR \wedge CLASS(v) = k\},$$

which means u^* is far from the "norm" in term of its distance to the cluster center, or

$$|\{v : v \in C_k \wedge d_k(u^*, v) \leq d_k(v, C_k)\}| \leq \alpha \cdot |C_k|,$$

which means u^* is amongst the most outreaching samples of the cluster.

Another popular definition of outlier is:

$$|\{v : v \in C_k \wedge d_k(u^*, v) \geq D\}| \leq \alpha \cdot |C_k|,$$

which means at least a fraction α of objects in C_k lies in a greater distance than D from u^*.

It can be observed that both approaches pay little attention to the problem of eliciting intensional knowledge from outliers, meaning no elaborated information that may help explain the reasons why a sample is considered outlier. This kind of knowledge is important for the validity evaluation of identified outliers, and certainly is useful in improving the overall understanding of the data.

Knorr and Ng made an attempt to address this issue by introducing the notion strength of outliers, derived from an analysis of dynamicity of outlier sets against changes in the features' subsets [26, 27]. Such analyzes belong to the very well established application domain of Rough Sets, and indeed a formalization of a similar approach within the framework of Rough Sets has been proposed by [23].

Our approach to outlier detection and analysis will assume a somewhat different perspective. It focuses on two main issues:

1. Elicitation of intensional knowledge from outliers by approximating the perception of external human experts.
2. Evaluation of suspicious samples by verification the performance of classifiers constructed using knowledge elicited from these samples.

Having established a mechanism for eliciting expert's knowledge as described in previous sections, we can develop outlier detection tests that might be completely independent from the existing similarity measures within the learning system as outlined in the Fig. 7 below:

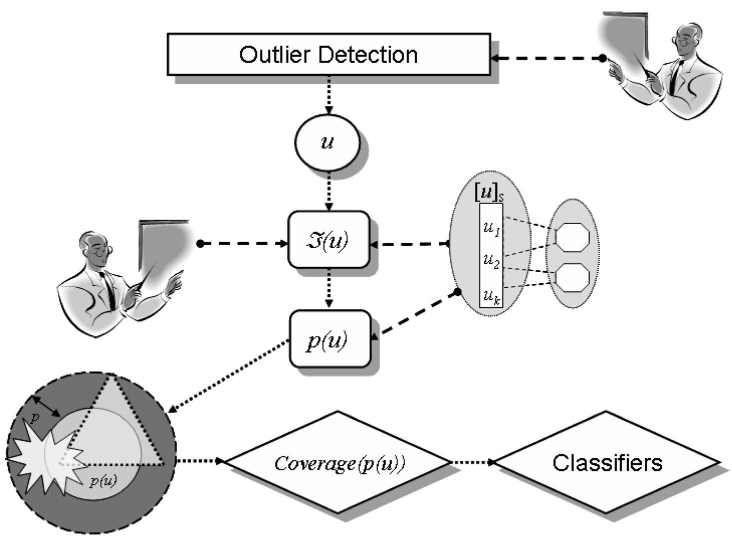

Fig. 7. Outlier analysis scheme

For a given training sample u^*,

Step 1. We ask the expert for his explanation on u^*. **Step 2.** The expert provides a foreign knowledge structure $\Im(u^*)$. **Step 3.** We approximate $\Im(u^*)$ under restrictive matching degrees to ensure only the immediate neighborhood of u^* is investigated. Let's say the result of such an approximation is a pattern (or set of patterns) p_u^*. **Step 4.** It is now sufficient to check $Coverage(p_u^*)$. If this coverage is high, it signifies that u^* may bear significant information that is also found in many other samples. The sample u^* therefore cannot be regarded as an outlier despite the fact that there may not be many other samples in its vicinity in terms of existing domestic distance measures of the learning system.

This test shows that distance-based outlier analysis and expert's elicited knowledge are complementary to each other.

In our architecture, outliers may be detected as samples that defied previous classification efforts, or samples that pass the above described outlier test, but may also be selected by the expert himself. This helps the classification system to focus on difficult samples in order to gradually improve the overall performance, in a way similar to that of popular boosting or leveraging algorithms. The main difference is that boosting algorithms employ a priori formulas/strategies to adjust weights to positive and negative samples, whereas our approach relies on the domain knowledge elicited from the external expert. In this way, we can benefit from the best of both sources of knowledge.

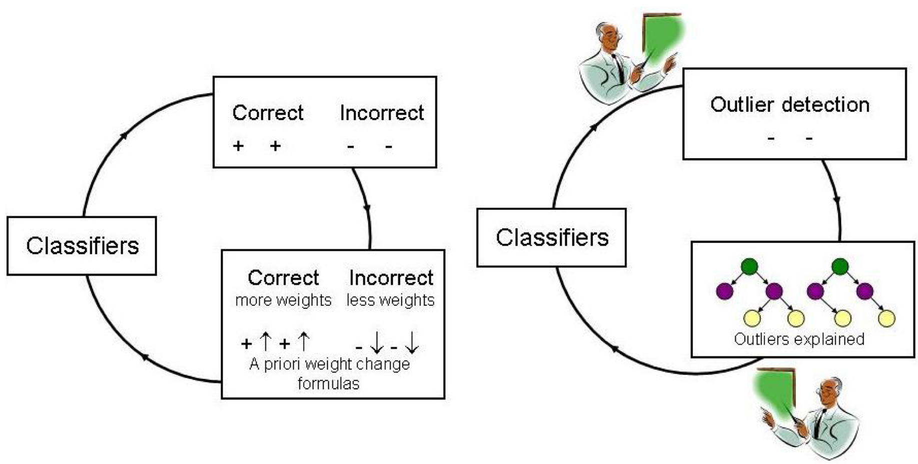

Fig. 8. Boosting vs Hierarchical Learning

2.6 Reinforcement Learning and Planning

The next two examples illustrate approximation of compound concepts in reinforcement learning and planning.

In reinforcement learning [13, 24, 29, 40, 43, 53, 58], 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. To approximate 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 $Q(s, a)$ [60]. We propose another approach to the approximation of $Q(s, a)$ based on ontology approximation. The approach is based on the assumption that in a dialog with experts 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. The explanation given by expert about possible values of $Q(s, a)$ may involve concepts from a special ontology. Using this ontology one can follow hierarchical learning methods to learn the approximations of its concepts. Such concepts can have a temporal character as well. This means the ranking of actions may depend not only on the current action and state but also on actions performed in the past, as well as on the changes caused by these actions.

In [6, 7] a computer tool based on rough sets for supporting automated planning of the medical treatment (see, e.g., [18, 59]) is discussed. In this approach, a given patient is treated as an investigated complex dynamical system, whilst diseases of this patient (RDS, PDA, sepsis, Ureaplasma and respiratory failure) are treated as compound objects changing and interacting over time. As a measure of planning success (or failure) in the experiments, we use a special hierarchical classifier that can predict the similarity between two plans as a number between 0.0 and 1.0. This classifier has been constructed on the basis of the special ontology specified by human experts and data sets. It is important to mention that in addition to the ontology, experts also provided the exemplary data (values of attributes) for the purpose of concepts approximation. The methods of construction such classifiers are based on approximate reasoning schemes (AR schemes, for short) and were described, e.g., in [5, 8, 9, 31]. We applied this method for approximation of similarity between plans generated in automated planning and plans proposed by human experts during the realistic clinical treatment.

2.7 Interaction with the Web

Let us discuss shortly problems which can be solved by human in dialog with the Web. Examples of such problems are considered in Service Oriented Computing or Service Oriented Architecture (see, e.g., [16]). Assuming that this dialog is performed in a simplified fragment of natural language [62, 64] one should develop tools for approximation concepts used in dialog to make them available to the Web for approximate reasoning in searching for the solutions of problems. One can expect that in a near future the Web can automatically help the users to synthesize required services if it will be possible to understood to satisfactory degree the specification received in dialog with users.

3 Selected Advanced Issues on GC

In this section, we discuss some advanced issues on GC. They are related to granules represented by agents or teams of agents interacting in changing environments.

We start from a general discussion on the *Wisdom technology* (wistech) system outlined recently in [21, 22].

Wisdom commonly means *rightly judging* based on available knowledge and interactions. This common notion can be refined. By *wisdom*, we understand an adaptive ability to make judgments correctly (in particular, correct decisions) to a satisfactory degree, having in mind real-life constraints. The intuitive nature of wisdom understood in this way can be metaphorically expressed by the so-called *wisdom equation* as shown in (4).

$$wisdom = adaptive\ judgment\ +\ knowledge\ +\ interaction. \qquad (4)$$

Wisdom can be treated as a certain type of knowledge. In particular, this type of knowledge is important at the highest level of hierarchy of meta-reasoning in intelligent agents.

Wistech is a collection of techniques aimed at the further advancement of technologies to acquire, represent, store, process, discover, communicate, and learn *wisdom* in designing and implementing intelligent systems. These techniques include approximate reasoning by agents or teams of agents about vague concepts concerning real-life, dynamically changing, usually distributed systems in which these agents are operating. Such systems consist of other autonomous agents operating in highly unpredictable environments and interacting with each others. Wistech can be treated as the successor of database technology, information management, and knowledge engineering technologies. Wistech is the combination of the technologies represented in equation (4) and offers an intuitive starting point for a variety of approaches to designing and implementing computational models for wistech in intelligent systems.

- *Knowledge technology* in wistech is based on techniques for reasoning about knowledge, information, and data, techniques that enable to employ the current knowledge in problem solving. This includes, e.g., extracting relevant fragments of knowledge from knowledge networks for making decisions or reasoning by analogy.
- *Judgment technology* in wistech is covering the representation of agent perception and adaptive judgment strategies based on results of perception of real life scenes in environments and their representations in the agent mind. The role of judgment is crucial, e.g., in adaptive planning relative to the Maslow Hierarchy of agents' needs or goals. Judgment also includes techniques used for perception, learning, analysis of perceived facts, and adaptive refinement of approximations of vague complex concepts (from different levels of concept hierarchies in real-life problem solving) applied in modeling interactions in dynamically changing environments (in which cooperating, communicating, and competing agents exist) under uncertain and insufficient knowledge or resources.
- *Interaction technology* includes techniques for performing and monitoring actions by agents and environments. Techniques for planning and controlling actions are derived from a combination of judgment technology and interaction technology.

The wistech system is strongly related to the idea of Gottfried Wilhelm Leibniz, one of the greatest mathematicians. He has discussed, in a sense, calculi of thoughts. In particular, he has written

> *If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, and say to each other: 'Let us calculate'.*
>
> – Gottfried Wilhelm Leibniz,
> Dissertio de Arte Combinatoria (Leipzig, 1666).

> *... Languages are the best mirror of the human mind, and that a precise analysis of the signification of words would tell us more than anything else about the operations of the understanding.*
>
> – Gottfried Wilhelm Leibniz,
> New Essays on Human Understanding (1705)
> Translated and edited by
> Peter Remnant and Jonathan Bennett
> Cambridge: Cambridge UP, 1982

Only much later, it was possible to recognize that new tools are necessary for developing such calculi, e.g., due to the necessity of reasoning under uncertainty about objects and (vague) concepts. Fuzzy set theory (Lotfi A. Zadeh, 1965) and rough set theory (Zdzisław Pawlak, 1982) represent two complementary approaches to vagueness. Fuzzy set theory addresses gradualness of knowledge, expressed by the fuzzy membership, whereas rough set theory addresses granularity of knowledge, expressed by the indiscernibility relation. Granular computing (Zadeh, 1973, 1998) may be now regarded as a unified framework for theories, methodologies and techniques for modeling of calculi of thoughts based on objects called granules.

There are many ways to build foundations for wistech computational models. One of them is based on the *rough-granular computing* (RGC) [52]. Rough-granular computing (RGC) is an approach for constructive definition of computations over objects called granules, aiming at searching for solutions of problems which are specified using vague concepts. Granules are obtained in a process called granulation. Granulation can be viewed as a human way of achieving data compression and it plays a key role in implementing the divide-and-conquer strategy in human problem-solving [64]. The proposed approach combines rough set methods with other soft computing methods, and methods based on granular computing (GC). RGC is used for developing one of the possible wistech foundations based on approximate reasoning about vague concepts.

Let us discuss some issues important to wistech, pertaining to compound granules which can perceive, and interact with, their environments.

3.1 Compound Granules in Perception and Interaction

Perception and interaction are closely related issues. Let us assume that an agent ag is perceiving the environment state e. The results of perceiving (e.g., by sensors) of different parts of e are stored in a a generalized information system IS (see Sect. 1.2). For each such a part s a partial information $Inf(s)$ is stored in IS together with information on relationships between parts. Form IS the structural model M_e of e is derived by hierarchical modeling. Granule M_e can be represented by a relational structure or a cluster of such structures). Next, the structural model M_e is matched against knowledge base network of the agent ag. The result of matching is a family $\mathcal{F}_e$ of concepts together with information about degrees to which these concepts are satisfied. Now, the judgment engine of ag is used to predict the current goals and to select of the relevant action (plan or communication) for required for interaction with the environment. Agent ag is attempting to make necessary changes or to move in the environment to reach the target goal. Note the environment changes are reflected by changes in the internal state representation of ag. Moreover, information $Inf(s')$, M'_s or $\mathcal{F}_{s'}$ about the next state e' of the environment is predicted, where s' denotes a part of e' perceived by ag. This information is further compared with the perceived information about the next state e', e.g., for reconstructing the current plan. The judgment engine of ag is also used for predicting changes of the internal state of ag caused by the actual environment state and by the actual internal state [30].

In [30] the above discussed judgment process was disussed in the framework of approximate reasoning about changes of the environment state and the agent state. In particular, this calls for approximation of complex functions characterizing these changes.

3.2 Coalition Formation and Interactive Computations

Coalitions of granules play an important role in interactive computations and reasoning about such computations. One can consider coalitions as operations on collections of granules representing agents to granules representing a meta-agent, i.e., a team of agents. Any such operation should provide for the construction of (i) the perception mechanism of coalition from the perception mechanisms of its members; (ii) the coalition judgment engine using the judgment engines of the coalition members; (iii) the coalition knowledge base network from knowledge base networks of the coalition members; (iv) the interaction mechanism of the coalition from the interaction mechanisms of the coalition members. For example, in a given situation, each member of coalition can have several choices for actions but the finally selected action by each member is based on cooperation of coalition members, i.e., they are choosing actions on the basis of, e.g., protocol of cooperation. Moreover, the environment is perceiving coalition as a whole, e.g., any action performed by coalition can be treated as the results of a vector $(a_1, \ldots, a_n)$ of actions of its members. These actions are selected on the basis of the accepted protocol. Note also perceived features of coalition refer to the coalition as the whole rather than its parts.

There are several challenging issues related to coalitions such as learning strategies for (hierarchical) coalition formation, constructing language of actions, plans, communication for coalition on the basis of languages of its members. Among these issues hierarchical coalitions play a special role in approximate reasoning about interactive computations, making it possible to perform approximate reasoning about the interaction of the whole system with the environment.

3.3 Granule Representation and Adaptation of Knowledge Base Networks

In the matching process of structural models with knowledge base networks, an important role play strategies of granule representations and geometry of granules [17].

Observe that during the perception, matching, and judgment processes the current knowledge base network may be updated. For example, some new patterns, concepts, production rules, rules, or approximate reasoning schemes are discovered and stored. Some other patterns may be removed from the knowledge base networks [48, 65].

3.4 Scalability Issues and Language Evolution

One can formulate several important questions related to languages used for communication by agents. Here are some examples. How the names for concepts are created? Why they are created? When we should go beyond ontologies of concepts? How the communication languages of agents are evolving when agents are dealing with the scalability issues, i.e., when the agents are trying to move from solving small size problems to large size problems?

4 Conclusions

We discussed the role of GC in approximation of complex concepts and in interactive computations, e.g., performed by agents interacting with environments and among themselves. We emphasized the role of dialogs of human experts with different granular systems in improving their performance. Finally, we formulated several challenges for approximate reasoning in granular systems interacting with the environments, in particular with other agents or human experts.

Acknowledgments. The research has been partially supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant “Decision support – new generation systems” of Innovative Economy Operational Programme 2008-2012 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

1. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. In: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp. 37–46. ACM Press, New York (2001)
2. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
3. Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley and Sons Ltd, Chichester (1978)
4. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press, Cambridge (1997)
5. Bazan, J.: Rough sets and granular computing in behavioral pattern identification and planning. In: Pedrycz, et al. (eds.) [39], pp. 777–800.
6. Bazan, J., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS, vol. 4259, pp. 418–427. Springer, Heidelberg (2006)
7. Bazan, J., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In: Proceedings of Information Processing and Management under Uncertainty in Knowledge-Based Systems (IPMU 2006), Paris, France, July 2-7, 2006, vol. (3), pp. 2650–2657. Editions E.D.K. (2006)
8. Bazan, J., Peters, J.F., Skowron, A.: Behavioral pattern identification through rough set modelling. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005. LNCS, vol. 3642, pp. 688–697. Springer, Heidelberg (2005)
9. Bazan, J., Skowron, A.: Classifiers based on approximate reasoning schemes. In: Dunin-Kęplicz, B., Jankowski, A., Skowron, A., Szczuka, M. (eds.) Monitoring, Security, and Rescue Techniques in Multiagent Systems. Advances in Soft Computing, pp. 191–202. Springer, Heidelberg (2005)
10. Bazan, J.G., Skowron, A., Świniarski, R.W.: Rough sets and vague concept approximation: From sample approximation to adaptive learning. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 39–62. Springer, Heidelberg (2006)
11. Behnke, S.: Hierarchical Neural Networks for Image Interpretation. LNCS, vol. 2766. Springer, Heidelberg (2003)
12. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Rec. 29(2), 93–104 (2000), http://portal.acm.org/citation.cfm?id=335191.335388
13. Dietterich, T.G.: Hierarchical reinforcement learning with the MAXQ value function decomposition. Artificial Intelligence 13, 227–303 (2000)
14. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Representation Techniques: A Rough Set Approach. Studies in Fuzziness and Soft Computing, vol. 202. Springer, Heidelberg (2006)
15. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, New York (2003)
16. Foster, I.: Service-oriented science. science 308, 814–817 (2005)
17. Gärdenfors, P.: The Geometry of Thought. MIT Press, Cambridge (2000)
18. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Elsevier, Morgan Kaufmann, San Francisco (2004)

19. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. 22(2), 85–126 (2004), http://dx.doi.org/10.1023/B:AIRE.0000045502.10941.a9
20. Jankowski, A., Peters, J., Skowron, A., Stepaniuk, J.: Optimization in discovery of compound granules. Fundamenta Informaticae 85(1-4), 249–265 (2008)
21. Jankowski, A., Skowron, A.: A wistech paradigm for intelligent systems. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 94–132. Springer, Heidelberg (2007)
22. Jankowski, A., Skowron, A.: Logic for artificial intelligence: The Rasiowa-Pawlak school perspective. In: Ehrenfeucht, A., Marek, V., Srebrny, M. (eds.) Andrzej Mostowski and Foundational Studies, pp. 106–143. IOS Press, Amsterdam (2008)
23. Jiang, F., Sui, Y., Cao, C.: Outlier detection using rough set theory. In: Slezak, D., Wang, M.D.I., Szczuka, G., Yao, Y. (eds.) 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Regina, Canada, pp. 79–87 (2005)
24. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 227–303 (1996)
25. Kleinberg, J., Papadimitriou, C., Raghavan, P.: A microeconomic view of data mining. Data Mining and Knowledge Discovery 2, 311–324 (1998)
26. Knorr, E.M.: Outliers and data mining: Finding exceptions in data. Ph.D. thesis, University of British Columbia (2002)
27. Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: VLDB 1999: Proceedings of the 25th International Conference on Very Large Data Bases, pp. 211–222. Morgan Kaufmann, San Francisco (1999), http://portal.acm.org/citation.cfm?id=671529
28. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: Algorithms and applications. The VLDB Journal 8(3), 237–253 (2000)
29. McGovern, A.: Autonomous Discovery of Temporal Abstractions from Interaction with an Environment. University of Massachusetts, Amherst, MA, PhD thesis (2002)
30. Nguyen, H.S., Jankowski, A., Skowron, A., Stepaniuk, J., Szczuka, M.: Discovery of process models from data and domain knowledge: A rough-granular approach. In: Yao, J.T. (ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, pp. 1–30. IGI Global, Hershey, PA (submitted, 2008)
31. Nguyen, S.H., Bazan, J., Skowron, A., Nguyen, H.S.: Layered learning for concept synthesis. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 187–208. Springer, Heidelberg (2004)
32. Nguyen, T.T.: Eliciting domain knowledge in handwritten digit recognition. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 762–767. Springer, Heidelberg (2005)
33. Nguyen, T.T., Paddon, C.P.W.D.J., Nguyen, S.H., Nguyen, H.S.: Learning sunspot classification. Fundamenta Informaticae 72(1-3), 295–309 (2006)
34. Oliveira, L., Sabourin, R., Bortolozzi, F., Suen, C.: Feature selection using multiobjective genetic algorithms for handwritten digit recognition. In: Proc. 16th International Conference on Pattern Recognition (ICPR 2002), pp. 568–571. IEEE Computer Society Press, Los Alamitos (2002)
35. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Berlin (2004)

36. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
37. Pawlak, Z., Skowron, A.: Rudiments of rough sets; Rough sets: Some extensions; Rough sets and boolean reasoning. Information Sciences 177(1), 3–27, 28–40, 41–73 (2007)
38. Pedrycz, W. (ed.): Granular Computing. Physica-Verlag, Heidelberg (2001)
39. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, New York (2008)
40. Peters, J.F.: Rough ethology: Towards a biologically-inspired study of collective behavior in intelligent systems with approximation spaces. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 153–174. Springer, Heidelberg (2004)
41. Peters, J.F.: Near sets: General theory about nearness of objects. Applied Mathematical Sciences 1(53), 2609–2629 (2007)
42. Peters, J.F.: Discovery of perceptually near information granules. In: J. Yao (ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation. Information Science Reference, Hersey, NY (to appear, 2008)
43. Peters, J.F., Henry, C.: Reinforcement learning with approximation spaces. Fundamenta Informaticae 71, 323–349 (2006)
44. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
45. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. International Journal of Approximate Reasoning 51, 333–365 (1996)
46. Polkowski, L., Skowron, A.: Constructing rough mereological granules of classifying rules and classifying algorithms. In: Bouchon-Meunier, B., Gutierrez-Rios, J., Magdalena, L., Yager, R.R., Kacprzyk, J. (eds.) Technologies for Constructing Intelligent Systems I, pp. 57–70. Physica-Verlag (2002)
47. Rissanen, J.: Minimum-description-length principle. In: Kotz, S., Johnson, N. (eds.) Encyclopedia of Statistical Sciences, pp. 523–527. John Wiley & Sons, New York (1985)
48. Skowron, A.: Approximate reasoning in distributed environments. In: Zhong, Liu (eds.) [65], pp. 433–474
49. Skowron, A.: Rough sets in perception-based computing. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 21–29. Springer, Heidelberg (2005)
50. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
51. Skowron, A.: Information granules and rough-neural computing. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies, pp. 43–84. Springer, Heidelberg (2003)
52. Skowron, A., Stepaniuk, J.: Rough sets and granular computing: Toward rough-granular computing. In: Pedrycz, et al. (eds.) [39], pp. 425–448.
53. Skowron, A., Stepaniuk, J., Peters, J., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)
54. Skowron, A., Stepaniuk, J., Peters, J.F.: Rough sets and infomorphisms: Towards approximation of relations in distributed environments. Fundamenta Informaticae 54(2–3), 263–277 (2003)

55. Skowron, A., Synak, P.: Complex patterns. Fundamenta Informaticae 60(1-4), 351–366 (2004)
56. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae 53(3–4), 365–387 (2002)
57. Stone, P.: Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. The MIT Press, Cambridge (2000)
58. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
59. Van Wezel, W., Jorna, R., Meystel, A.: Automated Planning: Theory and Practice. John Wiley & Sons, Chichester (2006)
60. Vapnik, V.: Statisctical Learning Theory. John Wiley & Sons, Chichester (1998)
61. Vapnik, V.N.: The Nature of Statistical Learning Theory (Information Science and Statistics). Springer, Heidelberg (1999)
62. Zadeh, L.A.: A new direction in AI - toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)
63. Zadeh, L.A.: From imprecise to granular probabilities. Fuzzy Sets and Systems 154(3), 370–374 (2005)
64. Zadeh, L.A.: Generalized theory of uncertainty (GTU)-principal concepts and ideas. Computational Statistics and Data Analysis 51, 15–46 (2006)
65. Zhong, N., Liu, J. (eds.): Intelligent Technologies for Information Analysis. Springer, Heidelberg (2004)