Adaptive aspects of combining approximation spaces

Jakub Wróblewski

Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
E-mail: jakubw@mimuw.edu.pl, http://alfa.mimuw.edu.pl/~jakubw/

Abstract. The paper addresses issues concerning a problem of constructing optimal classification algorithm. A notion of parameterized approximation space is used to model a process of the classifier construction. The process can be viewed as hierarchical searching for optimal information granulation to fit a concept described by empirical data. A problem of combining several parameterized information granules (given by classification algorithms) to obtain global data description is described. Some solutions based on adaptive methods are presented.

1 Introduction

Many practical, complex problems cannot be solved efficiently (because of e.g. computational limitations) without any form of decomposition of them into easier subproblems. A hierarchical approach to problem solving is widely known and used, as in case of control problem (*layered learning* [32]) or decomposition of large databases in KDD (knowledge discovery in databases) [10]. Granular computing [36] [24] [12] (a new paradigm in computer science, based on the notion of information granulation) as a machine learning, machine perception and KDD tool, also utilizes advantages of hierarchical structure.

The paper addresses issues concerning a problem of constructing optimal classification algorithm in KDD applications. Suppose that data is stored within *decision tables* [14], where each training case (elementary information granule) drops into one of predefined decision classes. By assumption, all available information about the universe of objects (cases) is collected in the decision table (or *information system*) $\mathbb{A} = (U, A, d)$, where each attribute $a \in A$ is identified with function $a : U \to V_a$ from the universe of objects U into the set V_a of all possible values on a and values $v_d \in V_d$ of $d \notin A$ (a distinguished decision attribute) correspond to mutually disjoint decision classes of objects. We will denote these classes by $D_1, ..., D_k$, where $D_i \subseteq U$.

The aim of data analysis is to construct a classifier (an algorithm which is able to classify previously unseen objects as members of appropriate decision classes) or an understandable description of data. Methods of construction of classifiers or descriptions can be regarded as tools for the data generalization, i.e. constructing more and more general description in terms of a hierarchy of

information granules. Classifiers based on the rough set theory [14] [17], are considered in the paper.

The main notion of the rough set theory is *indiscernibility relation*. Any two objects $u_1, u_2 \in U$ are indiscernible by a set of attributes $B \subseteq A$ (which is denoted by $(u_1, u_2) \in IND(B)$) iff there is no attribute $b \in B$ such that $b(u_1) \neq b(u_2)$. Indiscernibility class of object $u \in U$ is the set of objects (denoted as $[u]_B$) indiscernible with u:

$$[u]_B = \{u' \in U \,:\, \forall_{b \in B}\, b(u) = b(u')\}$$

A *decision reduct* $B \subseteq A$ is the minimal (in terms of inclusion) set of attributes which is sufficient to discern any pair of objects from different decision classes (supposing the whole set of attributes discerns this pair): $IND(B) \subseteq IND(\{d\}) \cup IND(A)$. Let us define the following rough set based notions:

Definition 1 *Let indiscernibility relation $IND(B)$ be given.* ***Upper approximation*** *of a set X is defined as:*

$$\overline{X} = \{u \in U :\, X \cap [u]_B \neq \emptyset\}$$

Lower approximation *of a set X is defined as:*

$$\underline{X} = \{u \in U :\, [u]_B \subseteq X\}$$

Definition 2 ***Rough inclusion*** *of a set Y in X is defined as:*

$$\mu(Y, X) = \begin{cases} \frac{|X \cap Y|}{|Y|} & if\ Y \neq \emptyset \\ 1 & otherwise \end{cases}$$

Rough membership *of object x in a set X basing on a set of attributes B is defined as :*

$$\mu_X^B(x) = \frac{|X \cap [x]_B|}{|[x]_B|}$$

Indiscernibility classes are related to different levels of information granulation. Elementary granules correspond to $[u]_A$ classes (based on the whole set of attributes), every $B \subset A$ corresponds to higher-level granule, which may be used as a base for decision rule:

$$a_1(u) = v_1 \wedge ... \wedge a_j(u) = v_j \Longrightarrow d(u) = v_d \tag{1}$$

for $B = \{a_1, ..., a_j\}$.

A notion of *approximation space*, a theoretical tool for data description with information granules is presented in the next sections of the paper. A general composition scheme of data models (regarded as approximation spaces) into one classifier is presented as well.

The reader can find more details on the important role of approximation spaces in the process of information granule construction in Chapter **??**.

2 Classification algorithms

2.1 Approximation spaces

A notion of *approximation space* (see, e.g., [21], [15], [22], [23], [25], [26], [27], [4]) may be regarded as an extension of the rough set theory. It is a tool for describing concepts not only in terms of their approximations, but also in terms of similarity of objects and concepts (see e.g., [15], [25], [23]). The notion of approximation space defined below is an extended form of definitions known from the literature (for more information see also Chapter **??** and [20]).

Definition 3 ***Approximation space*** *is a tuple* $AS = (U, I, \mathcal{R}, \nu)$*, where:*

U *– a set of objects;*

$I : U \longrightarrow \mathcal{P}(U)$ *– a function mapping every object from* U *into a subset (called* ***neighborhood****), where* $\forall_{u \in U} \quad u \in I(u)$

$\mathcal{R} \subseteq \mathcal{P}(U)$ *– a family of subsets of* U *(interpreted as a* ***set of templates****, or information granules, which are used to describe a concept)*

$\nu : \mathcal{P}(U) \times \mathcal{P}(U) \longrightarrow [0, 1]$ *– a function (interpreted as a* ***degree of inclusion*** *of subsets of* U*), where (cf. [26], [23]):*

1. $\forall_{A \subseteq U}\ \nu(A, A) = 1$
2. $\forall_{A \subseteq U}\ \nu(\emptyset, A) = 1$
3. $\forall_{A,B,C \subseteq U}\ \nu(A, B) = 1 \Rightarrow\ \nu(C, B) \geq \nu(C, A)$

An approximation space determines a language of describing concepts in U. It is useful especially in case of vague, inaccurate and incomplete description of data. Function I expresses an idea of indiscernibility of objects (a result of incompleteness of objects' description), whereas family $\mathcal{R}$ determines a way of generalizing information about objects (which allows us to deal with inaccurate and vague data). $\mathcal{R}$ may be defined e.g. by using language L of formulas basing on descriptors $a_i(u) = v$ as atomic formulas (for $a \in A$, $v \in V_a$) and operation "$\wedge$". In this case [27]:

$$\mathcal{R}_L = \{r_\alpha\ :\ \alpha \in L\} \tag{2}$$

where $r_\alpha \subseteq U$ corresponds to semantics of formula α in set U.

A goal of KDD process in both descriptive and predictive sense is to provide the best approximation of (one or more) concept $D \subset U$ basing on a known data, by optimal information granulation. In case of prediction task the approximation takes a form of a *classification algorithm* – a function mapping vectors of values of conditional attributes into the set of decision classes $\{D_1, ..., D_k\}$. Selected decision class $D_i \subseteq U$ is described by AS as a rough set with upper and lower approximation given by:

$$\overline{D_i} = \bigcup_{R \in \mathcal{R}\ :\ R \cap D_i \neq \emptyset} R$$

$$\underline{D_i} = \bigcup_{R \in \mathcal{R}\ :\ R \subseteq D_i} R$$

Definition 4 *Let $\mathbb{A}_1 = (U_1, A, d)$ be a decision table (training data set) and $AS = (U, I, \mathcal{R}, \nu)$ be an approximation space, where $U_1 \subseteq U$. Let $D \subseteq \mathcal{P}(U)$ be a partition of U onto disjoint decision classes $D = \{D_1, ..., D_k\}$, and let functions:*

$$\varrho : \mathcal{R} \to \{\emptyset, 1, 2, ..., k\}$$

where $k = |D|$ and:

$$\Phi : (\{\emptyset, 1, ..., k\} \times [0, 1])^* \longrightarrow \{\emptyset, 1, ..., k\}$$

be given. ***Classification algorithm*** *based on AS and ϱ, Φ is a mapping:*

$$CA_{AS,D,\varrho,\Phi} : U \longrightarrow \{\emptyset, D_1, D_2, ..., D_k\}$$

defined as:

$$CA_{AS,D,\varrho,\Phi}(u) = \Phi\left((\varrho(R_1), \nu(I(u), R_1)), ..., (\varrho(R_n), \nu(I(u), R_n))\right) \tag{3}$$

where $n = |\mathcal{R}|$. (We will omit subscripts AS, D, ϱ, Φ for simplicity).

Typically, a given test object u is matched against templates from family $\mathcal{R}$ (e.g. the left hand sides of decision rules) and the best matching $R \in \mathcal{R}$ is selected. Then the most frequent decision class in R is taken as a result of classification of u. In most cases ϱ is defined as:

$$\varrho(R) = \begin{cases} argmax_{i=1..k}(\nu(R, D_i)) & for\ \max_{i=1..k}(\nu(R, D_i)) > 0 \\ \emptyset & otherwise \end{cases} \tag{4}$$

If an object can be matched to more than one template R, the final answer is selected by voting:

$$\Phi((v_1, x_1), ..., (v_n, x_n)) = \begin{cases} argmax_{i=1..k}(\sum_{j \leq n:\, v_j = i} x_j) & if\ \exists_j x_j > 0 \\ \emptyset & if\ \forall_j\, x_j = 0 \end{cases} \tag{5}$$

for $n = |\mathcal{R}|$, i.e. given a set of partial answers v_i and corresponding coefficients x_i one should select the most popular answer (in terms of the sum of x_i). The coefficients may be regarded as support of decision, credibility or conviction factor etc. In case of formula 3 it is the coefficient of relevancy of template R_i, i.e. degree of inclusion of the test object in R_i.

Given template R may belong to upper approximation of more than one decision class. The conflict is resolved by function ϱ. Alternatively, the definition of classification algorithm may be extended onto sets of decision classes or even onto probability distributions over them:

$$CA : U \longrightarrow \Delta^k$$

where Δ^k denotes k-dimensional simplex: $\Delta^k = \{x \in [0, 1]^k \ : \ \sum_{i=1}^k x_i = 1\}$. In more general case the classification algorithm may take into account a degree of inclusion of an object u in a template R as well as inclusion of R in decision classes.

2.2 Parameterized approximation spaces

The notion of a ***parameterized approximation space*** was introduced [18] [35] to provide more flexible, data-dependent description language of the set U. By AS_ξ we will denote[1] an approximation space parameterized with a parameter vector $\xi \in \Xi$. The problem of optimal classifier construction is regarded as an optimization problem of finding optimal $\hat{\xi} \in \Xi$, i.e. of finding a vector of parameters such that $AS_{\hat{\xi}}$ generates optimal (in a sense of e.g. cross-validation results) classification algorithm. Parameter ξ is often used to maintain a balance between generality of model (classifier) and its accuracy.

Example 1 *Approximation space based on the set of attributes $B \subseteq A$ of information system $\mathbb{A} = (U, A, d)$ (cf. [26]). Let:*

$$\begin{aligned} &I(u) = [u]_A \\ &\mathcal{R} = \{[u]_B \,:\, u \in U\} \\ &\nu(X_1, X_2) = \mu(X_1, X_2) \end{aligned}$$

for $X_1, X_2 \subseteq U$, where μ is rough inclusion function (def. 2). Then $AS = (U, I, \mathcal{R}, \nu)$ is an approximation space related to a partition of set U into indiscernibility classes of relation $IND_{\mathbb{A}}(B)$. If we assume that B is a decision reduct of consistent data table $\mathbb{A}$, then the family $\mathcal{R}$ corresponds to a set of consistent decision rules (i.e. for all $R \in \mathcal{R}$ there is a decision class D_i such that $R \subseteq D_i$). Every template $R \in \mathcal{R}$ corresponds to a decision rule r of the form of conjunction of $a_i(u) = v_j$ descriptors, where $a_i \in B$, $v_j \in V_{a_i}$.

Now, let $AS_{B,\alpha}$ where $B \subseteq A$ and $\alpha \in [0, 1]$, be a parameterized approximation space defined as follows (cf. [37] [39]):

$$\begin{aligned} &I(u) = [u]_A \\ &\mathcal{R} = \{[u]_B \,:\, u \in U\} \\ &\nu(X_1, X_2) = \begin{cases} \mu(X_1, X_2) & if\ \mu(X_1, X_2) \geq \alpha \\ 0 & otherwise \end{cases} \end{aligned}$$

A classification algorithm based on $AS_{B,\alpha}$ works as follows: for any test object $u \in U$, find a template R matching it (i.e. a class of training objects identical to u with respect to attributes B), then check which is the most frequent decision class in a set R. If the most frequent decision class D_i covers at least α of R (i.e. $\mu(R, D_i) \geq \alpha$), object u is classified as the member of D_i (i.e. $\varrho(R) = i$). Otherwise, it is unclassified.

A goal of the above rough set based adaptive classification algorithm is to find such parameters (B, α) that the approximation space $AS_{B,\alpha}$ generates the best classifier. One can see, that with parameter B we adjust generality of the model (the smaller B is, the more general set of rules is generated, but also the less accurate rules we obtain). On the other hand, parameter α adjusts a

[1] The notion of parameterized approximation space is regarded in literature as $AS_{\$,\sharp} = (U, I_\$, \nu_\sharp)$. Notation used in this paper is an extension of the classical case.

degree of credibility of obtained model: for $\alpha = 1$ there may be many unclassified objects, but only credible rules are taken into account; for small α there may be no unclassified objects, but more objects are misclassified.

Example 2 *Let ρ be a metric over the set of objects U divided onto disjoint decision classes $D = \{D_1, ..., D_m\}$. For each $u \in U$ and for test data set U_1 let $\sigma_{u,\rho}$ be a permutation of $\{1, .., |U_1|\}$, such that:*

$$1 \leq i \leq j \leq |U_1| \Leftrightarrow \rho(u, u_{\sigma_{u,\rho}(i)}) \leq \rho(u, u_{\sigma_{u,\rho}(j)})$$

for $u_{\sigma_{u,\rho}(i)}, u_{\sigma_{u,\rho}(j)} \in U_1$.

Let $kNN_\rho : U \times \mathbb{N} \to 2^{U_1}$ be a function mapping each object u to a set of its k nearest neighbors according to metric ρ:

$$kNN_\rho(u, k) = \left\{u_{\sigma_u(1)}, ..., u_{\sigma_u(k)}\right\}$$

Let $I_{k,\rho}(u) = kNN_\rho(u, k)$ for a given k, let $\mathcal{R} = \{R \subseteq U : |R| = k\}$ and $\nu(X_1, X_2) = \mu(X_1, X_2)$ (cf. definition 2). Assume that ϱ and Φ are defined by equations 4 and 5. Then $AS = (U, I_k, \mathcal{R}, \nu)$ is approximation space and $CA_{AS,D,\varrho,\Phi}$ is a classification algorithm identical with the classical k-nearest neighbors algorithm. For each test object u we check its distance (given by a metric ρ) to all training objects from U_1. Then we find k nearest neighbors (set $I_{k,\rho}(u)$) and define template $R = I_k(u)$. Object u is then classified to the most frequent decision class in R.

Let $n = |A|$ and $w \in \mathbb{R}^m$. Let ρ_w be the following metric:

$$\rho_w(u_1, u_2) = \sum_{i=1}^{n} w_i |a_i(u_1) - a_i(u_2)|$$

The approximation space defined above may be regarded as the parameterized approximation space $AS_{k,w} = (U, I_{k,\rho_w}, \mathcal{R}, \nu)$, based on k nearest neighbors and metric ρ_w. It is known that proper selection of parameters (metric) is crucial for k-NN algorithm efficiency [2]).

3 Modeling classifiers as approximation spaces

Efficiency of classifier based on a given approximation space depends not only on domain-dependent information provided by values of attributes, but also on its granularity, i.e. level of data generalization. Proper granularity of attributes' values depends on knowledge representation (data description language) and generalization techniques used in classification algorithm. In case of data description by an approximation space $AS = (U, I, \mathcal{R}, \nu)$, the generalization is expressed by a family $\mathcal{R}$ of basic templates (granules) which form a final data model.

Some classification methods, especially these based on decision rules of the form 1, act better on discrete domains of attributes. Real-valued features are often transformed by discretization, hyperplanes, clustering, principal component

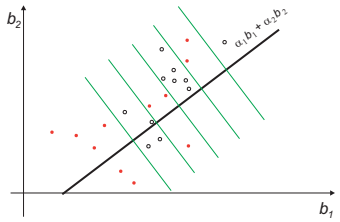

Fig. 1. Linear combination of two attributes and its discretization.

analysis etc. (cf. [6] [9] [11]). One can treat analysis process on transformed data either as a modeling of new data table (extended by new attributes given as a function of original ones) or, equivalently, as an extension of model language. The latter means, e.g. change of metric definition in k-NN algorithm (example 2) or extension of descriptor language in rule based system to interval descriptors "$a(u) \in [c_i, c_{i+1})$".

An example of a new attribute construction method was presented by the author in [29]. A subset of attributes $B = b_1, ..., b_m \subseteq A$ is selected, then an optimal (in a sense of some quality measure) linear combination of them is constructed by evolution strategy algorithm:

$$h(u) = \alpha_1 b_1(u) + ... + \alpha_m b_m(u)$$

where $\overrightarrow{\alpha} = (\alpha_1, ..., \alpha_m) \in \mathbb{R}^m$ is a vector of coefficients (assume $\|\overrightarrow{\alpha}\| = 1$). Note that every linear combination h corresponds to one vector of size $n = |A|$. An approximation space is based on a set of attributes containing a new one, being a discretization of h (see figure 1). If the process of constructing classification system involves extension of $\mathbb{A}$ with k new attributes based on linear combinations, one may regard the process as optimization of an approximation space $AS_{\xi,\overrightarrow{\alpha_1},...,\overrightarrow{\alpha_k}}$ parameterized by a set of parameters ξ (see example 1) and a set of vectors $\overrightarrow{\alpha_1}, ..., \overrightarrow{\alpha_k}$ representing linear combinations of attributes.

The more general approach is presented in [35]. A model based on a notion of relational information system [33], originally designed for relational databases analysis, can be easily extended to cover virtually all possible transformations of existing data. An inductive closure $\mathbb{A}^*$ of an information system (or a relational information system) $\mathbb{A}$ is a decision table closed by an operation of adding (inequivalent) new attributes based on a given family of operations. Such closure $\mathbb{A}^*$ is always finite since there is only finite number of inequivalent attributes of any decision table $\mathbb{A}$. Hence, any classifying system based on transformed attributes may be modeled by a parameterized approximation space $AS_{\xi,B}$, where ξ is a set of parameters (influencing e.g. a generalization level of rules) and $B \in A^*$ is a subset of attributes of inductive closure of $\mathbb{A}$.

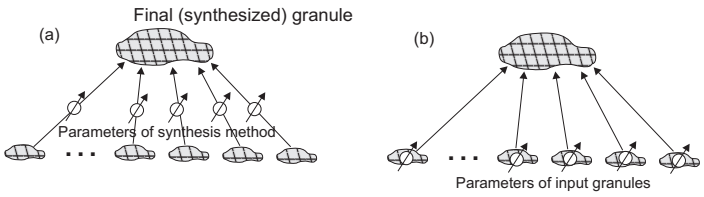

Fig. 2. Two general methods of adaptive combining granules: a) by weights, b) by adjusting model parameters on the lower level of synthesis tree.

When a final set of attributes (original, transformed or created basing on e.g. relations and tables in relational database) is fixed, the next phase of classifier construction begins: data reduction and model creation process. In a case of rough set based data analysis both these steps are done by calculation of reducts (exact or approximate – see [37] [28] [35] [31]) and a set of rules based on them. Unfortunately, the set of rules based on a reduct is too less general to provide good classification results. A combination of rule sets (classifiers), each of them based on a different reduct, different transformations of attributes and even on different subsets of training objects must be performed.

4 Combining approximation spaces

One may distinguish between two main adaptive methods of granules combination (see figure 2). The first one (denoted as "a)") is based on a vector of weights (real numbers) used in combination algorithm to adjust somehow the influence of a granule to a final model. In this case granules (given by classification algorithms) are fixed and the best vector of weights is used just to "mix" them (see the next section for more details). The second method (denoted as "b)") consists in changing parameters of input granules, e.g. their generality, for a fixed combining method. In this section we will consider one of the simplest adaptive combining method: by zero-one weights, which is equivalent to choosing a subset of classifiers and combining them in a fixed way. This subset we will refer to as "ensemble" of classifying agents (algorithms, represented by an approximation space).

Assume that a classification system CA is composed of k classifying agents, each of them based on its own parameterized approximation space AS_1, ..., AS_k and on its own subset of training examples $U_1, ..., U_k$ (using the same $I(u)$ and ν functions, limited to U_i). Let us define an approximation space being a combination of AS_1, ..., AS_k:

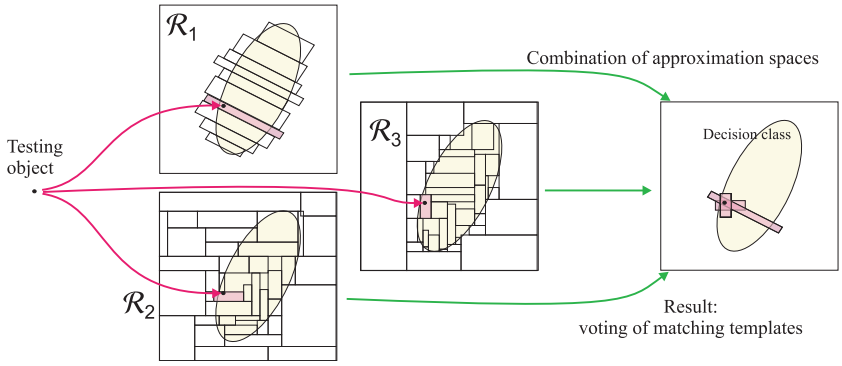

Fig. 3. Combination of approximation spaces (algorithms) and a new object classification.

Definition 5 ***Operation of synthesis*** *of approximation spaces AS_1, ..., AS_k, where $AS_i = (U_i, I_i, \mathcal{R}_i, \nu_i)$ and $I_i = I|_{U_i}$, $\nu_i = \nu|_{U_i}$, is a mapping S such that $S(AS_1, ..., AS_k) = AS'$, where $AS' = (U, I, \mathcal{R}, \nu)$ and:*

$$U = \bigcup_{i=1..k} U_i$$
$$\mathcal{R} = \bigcup_{i=1..k} \mathcal{R}_i$$

Classification of a new object u using AS' consists in finding all appropriate templates R (i.e. such R that $\nu(I(u), R)$ is large enough, see definitions 3 and 4). Then all values of $\varrho(R)$ are collected and the final answer is calculated by voting (function Φ).

Supposing that subsets U_i are significantly less than U, one can see that templates (in term of subsets of objects matched) $R_{i,j} \in \mathcal{R}_i$ are relatively small as well. In practice one should use a method of generalizing these templates onto the whole universe U. If, for example, a family $\mathcal{R}_i$ is defined by a reduct $B \subseteq A$ (see example 1):

$$\mathcal{R}_i = \{[u]_B \ : \ u \in U_i\}$$

then it will be generalized onto:

$$\mathcal{R}'_i = \{[u]_B \ : \ u \in U\}$$

and a definition of synthesized $S(AS_1, ..., AS_k) = AS'$ contains the following family $\mathcal{R}$:

$$\mathcal{R} = \bigcup_{i=1..k} \mathcal{R}'_i$$

In [35] some remarks concerning connections between above operations and rough mereology [16] are presented. A classification system based on a family of

approximation spaces may be regarded as a multiagent system with one special agent for result synthesis. In case of classification of a new object u, synthesizing agent sends to subordinate agents a request of delivery of partial descriptions (templates R) of object u. Then, a complete description is synthesized basing on definition 5.

Note that a set of classifying agents may work on a separate subsets $U_1, ..., U_n$ of set U (e.g. in distributed data mining system). Suppose that a set of approximation spaces AS_1, ..., AS_n was created basing on reducts (see example 1). Each AS_i is composed of a set of decision reducts, each of them related to one template $R \in \mathcal{R}_i$ (R is a set of objects matching the left hand side of the rule) and a decision value $d = \varrho(R)$. We tend to obtain the optimal synthesis of AS_1, ..., AS_n, basing on a measure Ψ of classification algorithm quality.

Let $S(AS_1, ..., AS_n) = AS'$, where $AS' = (U, I, \mathcal{R}, \nu)$. Suppose that:

$$\begin{array}{l} U = \bigcup_{i=1..n} U_i \\ \mathcal{R} = \bigcup_{i=1..n} \mathcal{R}_i \end{array}$$

for $AS_i = (U_i, I, \mathcal{R}_i, \nu)$. The space AS' is composed of all agents (approximation spaces) from the family AS_1, ..., AS_n; our goal is to choose such subset $J = \{j_1, ..., j_{|J|}\}$ that corresponds to synthesized approximation space:

$$AS_J = S(AS_{j_1}, ..., AS_{j_{|J|}}) \tag{6}$$

providing optimal classification algorithm CA_{AS_J}. Let $Pos_{\mathbb{B}}(CA)$ and $Neg_{\mathbb{B}}(CA)$ denote a number of testing objects from table $\mathbb{B}$ properly and improperly (respectively) classified by CA. Let Ψ be a **quality measure** based on classification results on $\mathbb{B}$, satisfying the following conditions:

$$\begin{array}{r} Pos_{\mathbb{B}}(CA_1) \subset Pos_{\mathbb{B}}(CA_2) \ \wedge \ Neg_{\mathbb{B}}(CA_1) = Neg_{\mathbb{B}}(CA_2) \ \Rightarrow \ \Psi(CA_1) < \Psi(CA_2) \\ Pos_{\mathbb{B}}(CA_1) = Pos_{\mathbb{B}}(CA_2) \ \wedge \ Neg_{\mathbb{B}}(CA_1) = Neg_{\mathbb{B}}(CA_2) \ \Rightarrow \\ \Rightarrow \ (\Psi(CA_1) < \Psi(CA_2) \Longleftrightarrow |J_1| > |J_2|) \end{array} \tag{7}$$

where $CA_1 = CA_{AS_{J_1}}$, $CA_2 = CA_{AS_{J_2}}$ and J_1, J_2 are subsets of agents. The above conditions mean, that if two subsets of agents achieve the same results on a test table $\mathbb{B}$, we would prefer the smaller one.

Assume that CA_{AS_J} is based on a voting function Φ such that:

$$(\forall_i \ v_i = v \vee v_i = \emptyset) \wedge (\exists_i \ v_i = v) \implies \Phi((v_1, 1), ..., (v_k, 1)) = v \tag{8}$$

The following fact is true for families of classifying agents (cf. [35]):

Theorem 1. *Let a quality function Ψ (meeting conditions 7) be given. Suppose AS_1, ..., AS_n are approximation spaces (classifying agents) based on reducts. A problem of finding optimal subset of agents (according to the function Ψ) is NP-hard.*

Proof: *A similar result (for a problem formulated in slightly different way) was presented in [34]. We will show that any minimal binary matrix column covering problem (known to be NP-hard) can be solved (in polynomial time) by selecting optimal subset of agents for a certain data table and set of classifying agents. Let* $\mathbf{B} = \{b_{ij}\}$ *be a* $n \times m$ *binary matrix to be covered by minimal set of columns (suppose there is at least one "1" in every row and column).*

Let $\mathbb{A} = (U, A, d)$ *be an information system such that every row of the matrix* $\mathbf{B}$ *corresponds to a pair of objects from* U*, every column of* $\mathbf{B}$ *corresponds to one attribute from* A *(hence* $|A| = n$*,* $|U| = 2m$*). Let attributes' values be defined as follows:*

$$\begin{array}{l} a_i(u_{2j-1}) = 2 - b_{ij} \\ a_i(u_{2j}) = 2 - 2b_{ij} \\ d(u_j) = j \bmod 2 \end{array}$$

where $j = 1..m$*,* $i = 1..n$*. The set* U *of objects is partitioned into two decision classes* D_0 *and* D_1*.*

Let us define a family of n *approximation spaces based on subtables:* $\mathbb{A}_i = (U_i, A, d)$*,* $i \in \{1, .., n\}$*, where* $U_i = \{u_{2j} \in U : b_{ij} = 1\} \cup \{u_{2j-1} \in U : b_{ij} = 1\}$*. Let* $AS_i = (U_i, I, \mathcal{R}_i, \nu)$ *be an approximation space based on the subtable* $\mathbb{A}_i$ *and subset of attributes* $B_i = \{a_i\}$ *(which is a reduct of* $\mathbb{A}_i$*):*

$$\begin{array}{l} I(u) = [u]_A \\ \mathcal{R}_i = \{[u]_{B_i} \ : \ u \in U_i\} \\ \nu(X_1, X_2) = \mu(X_1, X_2) \end{array}$$

The set U_i *contains these pairs of objects* u_{2j}, u_{2j-1} *which correspond to rows* $\mathbf{B}$ *covered by column* i*. Let* AS_J *be an approximation space based on* J *(equation 6). We will prove that classification algorithm* CA_{AS_J} *classifies correctly each object from* U *iff* J *corresponds to column covering of* $\mathbf{B}$*. Let* u_k *be an object from* U *(suppose, without loss of generality, that* k *is even,* $k = 2i$*). Let* $\mathcal{R}_J = \bigcup_{j \in J} \mathcal{R}_j$ *be a family of templates of synthesized approximation space* AS_J*. Note that for any* $R \in \mathcal{R}_j$*:*

$$u_{2i} \in R \in \mathcal{R}_j \iff b_{ij} = 1$$

hence, as J *corresponds to a covering of* $\mathbf{B}$*, there exists a template* R *which matches the object* u_k*. Note that for even numbers of objects:*

$$[u_{2i}]_{B_j} = D_0$$

where $u_{2i} \in U_j$*. Hence:*

$$u_{2i} \in R \in \mathcal{R}_j \implies \varrho(R) = 0$$

Every rule based on a template $R \in \mathcal{R}_J$ *is deterministic, therefore for any voting function* Φ *(which meets the condition 8) the object* u_k *will be classified correctly. The same holds for odd* k *(in this case* $\varrho(R) = 1$*).*

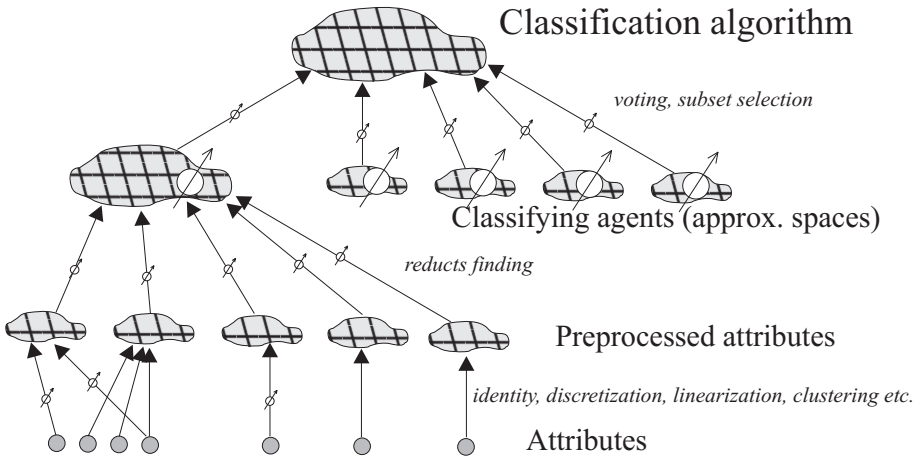

Fig. 4. Hierarchical construction of classifying algorithm from granules (descriptors, approximation spaces). Small circles with arrows denote adaptable parameters of information granules (or transforming/combining them).

Suppose that J corresponds to a set of columns which is not a covering of $\mathbf{B}$*. In this case there exists a row i not covered by any of selected columns and object u_{2i} is not contained by any U_j for $j \in J$. The object u_{2i} does not match any template from $\mathcal{R}_J$, so it will not be classified correctly.*

There was proven that there exists a bijection between ensembles (subsets) of classifying agents (which classifies correctly all objects from $\mathbb{A}$) and coverings (subsets of columns) of matrix $\mathbf{B}$*. Note that (by assumption 7) if there are many ensembles which classify every object in U, a function Ψ will prefer the smaller one. Hence optimal subset of agents corresponds to the minimal covering of* $\mathbf{B}$*. This completes a construction of (polynomial) transformation of matrix covering problem to the problem of selection of optimal subset of agents, which proves NP-hardness of the latter.*

5 Adaptive strategies of constructing classifiers

KDD process [5] consists of several stages; some of them may be performed automatically (some preprocessing steps, data reduction, method selection, data mining), whereas the others require an expert knowledge (understanding application domain, goals of analytic process, selecting an appropriate data set, interpreting and utilizing results). One of the important fields in KDD research is to develop methods of automatization of possibly many steps of KDD process, by using e.g. automatic feature extraction, data reduction or algorithm selection (via parameterization). These methods are often based on adaptation paradigm.

Let us consider an automatic classification system based on the KDD scheme. We will construct the classification algorithm step by step, by optimizing infor-

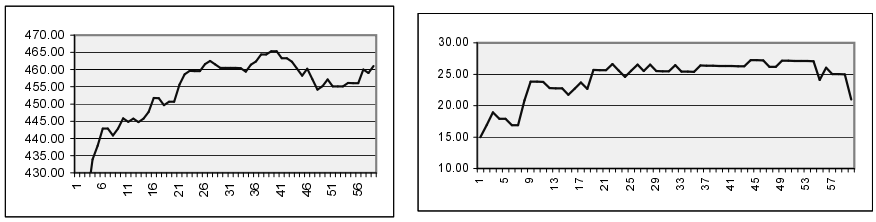

Fig. 5. Classification results (vertical axis) and number of agents in an ensemble (horizontal axis) – `DNA_splices` and `primary_tumor` data sets.

mation granulation used at each level: feature extraction and preprocessing, data reduction and generalization, synthesis of the final classifier (see figure 4). Some of these steps are known to be NP-hard, e.g. optimal decomposition problem [11], optimal reduct finding (in a sense of its length or other measures, also in case of approximate or dynamic reducts [31] [35]), selection of optimal ensembles of agents (see above and [34] [35]). Approximate adaptive heuristics (e.g. based on evolutionary metaheuristics) should be used to optimize these steps.

A practical (partial) implementation of a classification system described on figure 4 was presented by author in [35]. On the lower level (feature extraction) evolutionary algorithms are used to create optimal linearization of attributes or new features based on relational database (cf. section 3). The process may be regarded as an optimization of weights in case of linearization, or as a selection (by 0-1 weights) of the best new attribute from the inductive closure of database. There are another potential spaces of structures of new attributes, based on both supervised and unsupervised learning methods (clustering, PCA, discretization; feature extraction methods used in case of complex input objects: time series analyses, pattern recognition etc.), which match the general scheme (fig. 4).

The rough set based rule induction system is used at the generalization stage of the algorithm. A group of adaptation based evolutionary (hybrid) algorithms for the reduct finding create a complete approximation space (by providing a set of rules as a source of templates forming $\mathcal{R}$ family) parameterized by approximation coefficients in case of approximate reducts [30]. The reduct finding process can be regarded as an optimization of 0-1 coefficients used in combining elementary granules (based on single attributes) into more complex ones (described by the approximation space).

The next step in the hierarchy depicted on figure 4 is concerned with creating optimal ensembles of classifying agents. The problem is NP-hard (see theorem 1); results of practical experiments confirm that increasing the number of agents in ensemble does not necessarily lead to enhancing classification results (see figure 5 and [34]). In [35] a genetic algorithm is used to find an optimal subset of agents. Chromosomes (binary coding) represent subsets of agents and fitness function is calculated by basing on classification results on an additional testing subtable.

There are two main conditions to regard an algorithm as the adaptive one [1]: first, the algorithm should be parameterized (able to change itself); second, the criterion of parameters' optimization should be based on algorithm's efficiency. In the case of adaptive scheme presented above (figure 4) every level of the hierarchical granule combination process is parameterized – either by weights (adjusting the method of combining granules) or by granules' parameters. The optimization process of these parameters (e.g. fitness function in case of genetic algorithms) at each level is based on an approximation (estimation) of the final classifier performance. In some cases the estimation is based on results for an additional test sample (e.g. in case of optimization of ensemble of agents [34]), at other levels one should use more indirect approximation. In case of adaptive system described in [35] both new features (e.g. given by linearization) and reducts are optimized by a probabilistic-based quality measure (*predictive measure* [33]) estimating the final classifier quality indirectly. The popular criteria of the classifier optimization, based on the minimum description length principle [7], lead to even more indirect approximation.

One may notice an interesting analogy between figure 4 and neural networks [13] [18]. In the case of multilayer feed-forward artificial neural net a model of input-output dependency is built as a combination of a number of linear (parameterized) and nonlinear functions. The adaptation process (implemented e.g. as a backpropagation algorithm) is based on adjusting parameters (weights) basing on the model prediction error, propagated downward the net. There is no direct way to adopt this scheme to the general case of adaptive rule-based classifier since there is no general methods of error propagation known in the discrete case (although some heuristics are used in this case). The most universal (but time-consuming) adaptation scheme is to collect new cases together with the correct answers and to rebuild the whole classification system or just a part of it (e.g. a new ensemble of agents) using the new data.

6 Results and conclusions

The paper describes a general scheme of modeling a process of classification system construction using a notion of information granule. The process starts with a set of elementary information granules based on single attributes. The first level of adaptive process of classifier construction is a preprocessing of the initial attributes: discretization (which means generalization of several information granules into one), linearization (combining several attributes using optimal in some sense linear combination of them; the final information granule is a combination of a set of granules based on a set of attributes) and other techniques.

The next level of the hierarchical process is to combine information granules derived from the original attributes into approximation spaces – collections of information granules of higher order. Rough set theory is the tool for the generalization of descriptors (granules based on single attributes) onto the sets of rules.

Data	Size (training table)	k-NN	C4.5	Result
sat_image	4435×37	90.6	85.0	91.05
letter	15000×17	95.6	88.5	96.00
diabetes	768×9	67.6	73.0	73.30
breast_cancer	286×10	73.1	71.0	72.84
primary_tumor	339×18	42.2	40.0	39.43
Australian	690×15	81.9	84.5	86.34
vehicle	846×19	72.5	75.2	68.61
DNA_splices	2000×181	85.4	92.4	95.29
pendigits	7494×16	97.8		98.28

Table 1. Experimental results compared with two popular classifiers. Column "Result" contains a number (percent) of properly classified test objects.

The last level of the process described in the paper is to combine a set of information granules (sets of rules, classifying agents) into one classification system and to resolve conflicts between them. A problem of optimal subset of agents selection is proven to be NP-hard and a genetic algorithm is proposed to solve it approximately.

Since many of the problems concerning constructing and combining information granules are proven to be NP-hard, approximate heuristics should be used to obtain good results. The adaptive paradigm is the base of algorithms described in the paper. All the steps of granules' combination are parameterized and some algorithms of parameters' optimization are presented. Quality measures based on (estimated) efficiency of classification of new cases are proposed.

The adaptive classification system described above was partially implemented by author [35] [29] [34]. Results of experiments on some benchmark data tables are presented in table 6.

Further research is needed in many detailed aspects of the described process. A regular examination of adaptive strategies of parameters optimization (especially in case generalization parameters, not only weights) should be performed. Although many parts of the process are successfully implemented by author, there are still no experimental results concerning the whole, fully adaptive algorithm. An integration of some methods described in the paper with RSES – rough set based data analysis system [19] – is to be done in the near future.

Acknowledgements

This work was supported by the grant of Polish National Committee for Scientific Research (KBN) No. 8T11C02519.

References

1. Bäck T.: An Overview of Parameter Control Methods by Self-Adaptation in Evolutionary Algorithms. Fundamenta Informaticae **35** (1), IOS Press (1998) 51–66.

2. Bay S.D.: Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets. Proc. of the International Conference of the Machine Learning. Morgan Kaufmann Publishers, Madison, Wisc. (1998).
3. Bazan J.G., Nguyen H.S., Nguyen S.H, Synak P., Wróblewski J.: Rough Set Algorithms in Classification Problem. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds), Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems, Physica-Verlag (2000) 49–88.
4. Düntsch I., Gediga G.: Uncertainty measures of rough set prediction. Artificial Intelligence, vol. 106 (1998) 77–107.
5. Düntsch I., Gediga G., Nguyen H.S.: Rough set data analysis in the KDD process. Proc. of IPMU 2000, Madrid, Spain. Universidad Politecnica de Madrid (2000) vol. 1, 220–226.
6. Jolliffe I.T.: Principal Component Analysis. Springer-Verlag (1986).
7. Li M., Vitanyi P.: An introduction to Kolmogorov complexity and its applications. Springer-Verlag, New York, Berlin, Heidelberg (1993).
8. Lin T.Y., Wildberger A.M. (eds.): Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery. Simulation Councils, Inc., San Diego, CA (1995).
9. Liu H., Motoda H. (eds): Feature extraction, construction and selection – a data mining perspective. Kluwer Academic Publishers, Dordrecht (1998).
10. Nguyen S.H., Polkowski L., Skowron A., Synak P., Wrblewski J.: Searching of Approximate Description of Decision Classes. Proc. of The Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, RSFD'96, Tokyo (1996) 153–161.
11. Nguyen H.S.: Discretization of Real Value Attributes: Boolean Reasoning Approach. Ph.D. thesis. Institute of Mathematics, Warsaw University, Poland (1997).
12. Nguyen H.S., Skowron A., Stepaniuk J.: Granular Computing: A Rough Set Approach. Computational Intelligence **17**(**3**) (2001) 514–544.
13. Pal S.K., Pedrycz W., Skowron A., Swiniarski R. (eds.): Rough-Neuro Computing (special volume). Neurocomputing, vol. **36** (2001).
14. Pawlak Z.: Rough sets – Theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht (1991).
15. Polkowski L., Skowron A., Żytkow J.: Tolerance based rough sets. In: [8], 55–58.
16. Polkowski L., Skowron A.: Rough Mereological Foundations for Design, Analysis, Synthesis and Control in Distributed Systems. Information Sciences – An International Journal 104/1-2, Elsevier Science, New York (1998) 129–156.
17. Polkowski, L., Skowron, A. (eds): Rough Sets in Knowledge Discovery vol. **1**, **2**. Physica-Verlag, Heidelberg (1998).
18. Polkowski, L., Skowron, A. (eds): Rough-Neuro Computing. Proc. of RSCTC-2000, Banff, University of Regina, Regina, Saskatchewan (2000) 25–32.
19. RSES homepage – rough set based data analysis system: http://alfa.mimuw.edu.pl/~rses/
20. Skowron, A.: Approximation Spaces in Rough Neurocomputing. In: Hirano S., Inuiguhi M., Tsumoto S. (eds.), Rough Set Theory and Granular Computing, Physica-Verlag (to appear).
21. Skowron A., Stepaniuk J.: Approximation of relations. In: [38] 161–166.
22. Skowron A., Stepaniuk J.: Generalized Approximation Spaces. In: T.Y. Lin, A.M. Wildberger (eds.): Soft Computing. San Diego Simulation Councils, Inc. (1995) 18–21.
23. Skowron A., Stepaniuk J.: Tolerance Approximation Spaces. Fundamenta Informaticae **27**, IOS Press (1996) 245–253.

24. Skowron A., Stepaniuk J.: Information granule decomposition. Fundamenta Informaticae **47**, IOS Press (2001) 337–350.
25. Słowiński R., Vanderpooten D.: Similarity relation as a basis for rough approximations. Institute of Computer Science, Warsaw University of Technology, ICS Research Report **53/95** (1995); see also: P.P. Wang (ed.), Advances in Machine Intelligence & Soft–Computing, Bookwrights, Raleigh, NC (1997) 17–33.
26. Stepaniuk J.: Approximation Spaces, Reducts and Representatives. In: L. Polkowski, A. Skowron (eds.). Rough Sets in Knowledge Discovery 2. Physica Verlag, Heidelberg (1998) 109–126.
27. Stepaniuk J.: Knowledge discovery by application of rough set methods. In: L. Polkowski, T.Y. Lin, S. Tsumoto (eds.): Rough sets: New developments, Studies in Fuzziness and Soft-Computing, Physica-Verlag / Springer-Verlag, Heidelberg (2000).
28. Ślęzak, D.: Approximate reducts in decision tables. Proc. of IPMU'96. Granada, Spain (1996) vol. 3, 1159–1164.
29. Ślęzak D., Wróblewski J.: Classification algorithms based on linear combinations of features. Proc. of PKDD'99. Praga, Czech Republik, LNAI 1704, Springer, Heidelberg (1999) 548–553.
30. Ślęzak D., Wróblewski J.: Application of normalized decision measures to the new case classification. Proc. of RSCTC-2000, Banff, University of Regina, Regina, Saskatchewan (2000) 515–522.
31. Ślęzak, D.: Approximate decision reducts (In Polish). Ph.D. thesis. Institute of Mathematics, Warsaw University, Poland (2001).
32. Stone P.: Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. MIT Press (2000).
33. Wróblewski J.: Analyzing relational databases using rough set based methods. In: Proc. of IPMU'00. Madrid, Spain (2000) 256–262.
34. Wróblewski J.: Ensembles of classifiers based on approximate reducts. Proc. of CS&P 2000 Workshop, Informatik-Bericht Nr. 140, Humboldt-Universität zu Berlin (2000) vol. 2, 355–362. Extended and revised version also in: Fundamenta Informaticae **47** (3,4), IOS Press (2001) 351–360.
35. Wróblewski J.: Adaptive methods of object classification. Ph.D. thesis, Institute of Mathematics, Warsaw University (2001).
36. Zadeh L.A., Kacprzyk J. (eds.): Computing with Words in Information/Intelligent Systems, vol. 1–2, Physica–Verlag, Heidelberg (1999).
37. Ziarko W.: Variable Precision Rough Set Model. Journal of Computer and System Sciences **46** (1993) 39–59.
38. Ziarko W. (ed.): Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93). Workshops in Computing, Springer–Verlag & British Computer Society, London, Berlin (1994).
39. Ziarko W.: Approximation Region-Based Decision Tables. Proc. of RSCTC'98, Warsaw, Poland. Springer-Verlag (LNAI 1424), Berlin Heidelberg (1998) 178–185.