Chapter 6
Adaptive Aspects of Combining Approximation Spaces

Jakub Wróblewski

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

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

1 Introduction

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

This chapter addresses issues concerning the problem of constructing an 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 a function $a : U \to V_a$ from the universe of objects U in the set V_a of all possible values of 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 $D_1, \ldots, D_k$, where $D_i \subseteq U$.

The aim of data analysis is to construct an understandable description of data or a classifier (an algorithm that can classify previously unseen objects as members of appropriate decision classes). Methods of constructing of classifiers or descriptions

can be regarded as tools for data generalization, i.e., tools that construct more and more general descriptions in terms of a hierarchy of information granules. Classifiers based on the rough set theory [14–17] are considered in this chapter.

The main notion of the rough set theory is the *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 $b(u_1) \neq b(u_2)$. An 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 that is sufficient to discern any pair of objects from different decision classes, supposing that the whole set of attributes discerns the 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. *The upper approximation* of a set X is defined as

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

The lower approximation of a set X is defined by

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

Definition 2. *The rough inclusion* of set Y in X is defined by

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

The rough membership of object x in set X based on a set of attributes B is defined by

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

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 a higher level granule, which may be used as a base for decision rule:

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

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

A notion of *approximation space*, a theoretical tool for data description with information granules is presented in the next sections of this chapter. 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 Chap. 3.

2 Classification Algorithms

2.1 Approximation Spaces

The notion of an *approximation space* (see, e.g., [4, 15, 21–23, 25–27]) may be regarded as an extension of rough set theory. It is a tool for describing concepts not only in terms of their approximations but also in terms of the similarity of objects and concepts (see e.g., [15, 23, 25]). The notion of approximation space defined below is an extended form of definitions known from the literature (for more information see also Chap. 3 and [20]).

Definition 3. *An approximation space* is a tuple $AS = (U, I, \mathcal{R}, \nu)$, where
• U is a set of objects.
• $I : U \longrightarrow \mathcal{P}(U)$ is a function mapping every object from U into a subset (called a *neighborhood*), where $\forall_{u \in U}\ u \in I(u)$.
• $\mathcal{R} \subseteq \mathcal{P}(U)$ is a family of subsets of U (interpreted as a *set of templates*, or information granules, which are used to describe a concept).
• $\nu : \mathcal{P}(U) \times \mathcal{P}(U) \longrightarrow [0,1]$ is a function (interpreted as the *degree of inclusion* of subsets of U), where (see [23, 26])

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 cases of vague, inaccurate, and incomplete descriptions of data. Function I expresses the idea of the indiscernibility of objects (a result of incompleteness of object descriptions), 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 based 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 the semantics of formula α in set U.

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

$$\overline{D_i} = \bigcup_{R \in \mathcal{R} : R \cap D_i \neq \emptyset} R; \qquad \underline{D_i} = \bigcup_{R \in \mathcal{R} : R \subseteq D_i} R.$$

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 $D = \{D_1, \ldots, D_k\}$, and let functions

$$\rho : \mathcal{R} \to \{\emptyset, 1, 2, \ldots, k\}$$

where $k = |D|$ and

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

be given. *The classification algorithm* based on AS and ρ, Φ is a mapping

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

defined as

$$CA_{AS,D,\rho,\Phi}(u) = \Phi\{\rho(R_1), \nu[I(u), R_1], \ldots, \rho(R_n), \nu[I(u), R_n]\}, \tag{3}$$

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 the 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 the classification of u. In most cases, ρ is defined as

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

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

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

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. For formula 3, it is the coefficient of relevancy of template R_i, i.e., the degree of inclusion of the test object in R_i.

Given template R may belong to the upper approximation of more than one decision class. The conflict is resolved by function ρ. Alternatively, the definition of the 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 the k-dimensional simplex: $\Delta^k = \{x \in [0,1]^k : \sum_{i=1}^k x_i = 1\}$. In more general cases, the classification algorithm may take into account the degree of inclusion of an object u in the template R as well as the 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 an optimal (in the sense of, e.g., cross-validation results) classification algorithm. Parameter ξ is often used to maintain a balance between the generality of a model (classifier) and its accuracy.

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

$$\begin{aligned} &1.\ I(u) = [u]_A, \\ &2.\ \mathcal{R} = \{[u]_B : u \in U\}, \\ &3.\ \nu(X_1, X_2) = \mu(X_1, X_2), \end{aligned}$$

for $X_1, X_2 \subseteq U$, where μ is a rough inclusion function (Def. 2). Then $AS = (U, I, \mathcal{R}, \nu)$ is an approximation space related to a partition of the set U into indiscernibility classes of the relation $IND_{\mathbb{A}}(B)$. If we assume that B is a decision reduct of consistent data table $\mathbb{A}$, then 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 the 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 (see [37, 39]):

$$\begin{aligned} &1.\ I(u) = [u]_A, \\ &2.\ \mathcal{R} = \{[u]_B : u \in U\}, \\ &3.\ \nu(X_1, X_2) = \begin{cases} \mu(X_1, X_2) & \text{if } \mu(X_1, X_2) \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

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 a member of D_i [i.e., $\rho(R) = i$]. Otherwise, it is unclassified.

The 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 the generality of the model (the

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

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 the degree of credibility of the model obtained: 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 on a set of objects U divided into disjoint decision classes $D = \{D_1, \ldots, D_m\}$. 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:

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

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 (4) and (5). Then $AS = (U, I_k, \mathcal{R}, \nu)$ is an approximation space, and $CA_{AS,D,\rho,\Phi}$ is a classification algorithm identical to the classical k-nearest neighbors algorithm. For each test object u, we check its distance (given by metric ρ) to all training objects from U_1. Then we find the k nearest neighbors [set $I_{k,\rho}(u)$] and define template $R = I_k(u)$. Object u is then classified into 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 the k nearest neighbors and metric ρ_w. It is known that the proper selection of parameters (metric) is crucial for k-NN algorithm efficiency [2]).

3 Modeling Classifiers as Approximation Spaces

The efficiency of a 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 attribute values depends on the knowledge representation (data description language) and the generalization techniques used in the classification algorithm. In cases 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) that 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 analysis, etc. [6, 9, 11]. One can treat the analysis process on transformed data either as modeling of a 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 the k-NN algorithm (Example 2) or extension of descriptor language by interval descriptors "$a(u) \in [c_i, c_{i+1})$" in a rule based system.

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

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

where $\vec{\alpha} = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$ is a vector of coefficients (assume $\|\vec{\alpha}\| = 1$). 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 that is a discretization of h (see Fig. 1). If the process of constructing a 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 $AS_{\xi,\vec{\alpha}_1,\ldots,\vec{\alpha}_k}$ parameterized by a set of parameters ξ (see Example 1) and a set of vectors $\vec{\alpha}_1, \ldots, \vec{\alpha}_k$ representing linear combinations of attributes.

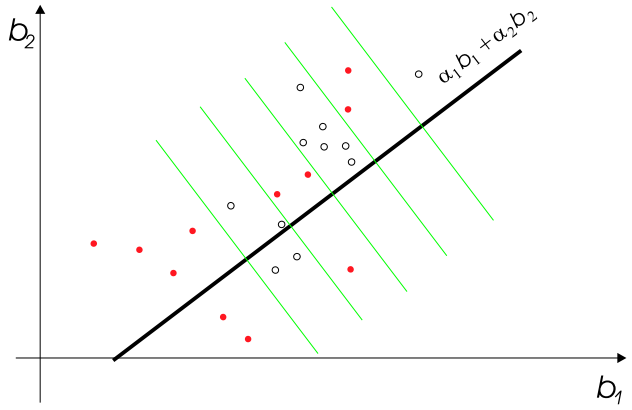

Fig. 1. A linear combination of two attributes and its discretization

The more general approach is presented in [35]. A model based on the notion of a relational information system [33], originally designed for relational database analy-

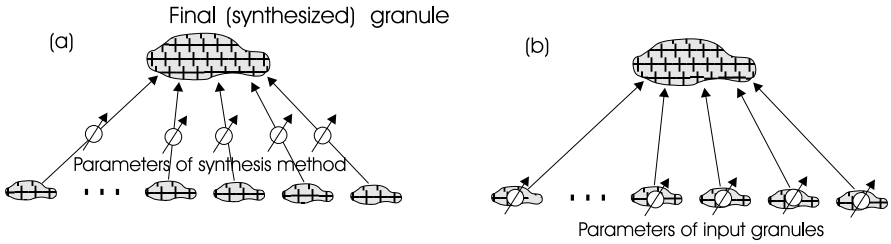

Fig. 2. Two general methods of adaptive combining granules: **(a)** by weights, **(b)** by adjusting model parameters on the lower level of a synthesis tree

sis, can be easily extended to cover virtually all possible transformations of existing data. The 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 a closure $\mathbb{A}^*$ is always finite since there is only a 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 $\mathbb{A}$.

When a final set of attributes (original, transformed, or created based on, e.g., relations and tables in relational database) is fixed, the next phase of classifier construction begins: data reduction and the model creation process. In rough set based data analysis, both steps are done by calculating reducts (exact or approximate) [28, 31, 35, 37] and a set of rules based on them. Unfortunately, a set of rules based on a reduct is not general enough 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 granule combination (see Fig. 2). The first denoted (a) is based on a vector of weights (real numbers) used in a combination algorithm to adjust, somehow, the influence of a granule on 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 (b) consists of 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 methods: by zero–one weights, which is equivalent to choosing a subset of classifiers and combining them in a fixed way. We will refer to this subset as an ensemble of classifying agents (algorithms represented by an approximation space).

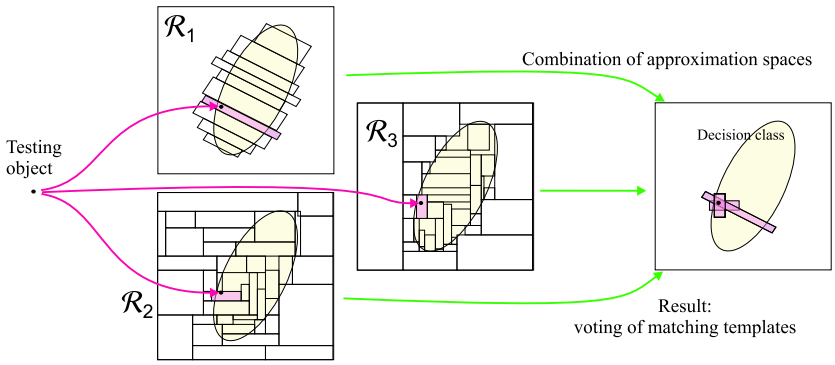

Fig. 3. A combination of approximation spaces (algorithms) and a new object classification

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

Definition 5. *Operation of synthesis* of approximation spaces $AS_1, \ldots, AS_k$, 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 $S(AS_1,\ldots,AS_k) = AS'$, where $AS' = (U, I, \mathcal{R}, \nu)$ and

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

The classification of a new object u using AS' consists of finding all appropriate templates R [i.e., R such that $\nu[I(u), R]$ is large enough, see Definitions 3 and 4]. Then all values of $\rho(R)$ 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 terms 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 $S(AS_1,\ldots,AS_k) = AS'$ contains the following family $\mathcal{R}$:

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

In [35], some remarks concerning connections between the 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. When classifying a new object u, the synthesizing agent sends a request for delivery of partial descriptions (templates R) of the object to subordinate agents. Then a complete description is synthesized based on Definition 5.

Note that a set of classifying agents may work on separate subsets $U_1,\ldots,U_n$ of set U (e.g., in a distributed data mining system). Suppose that a set of approximation spaces $AS_1,\ldots,AS_n$ was created based 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 $d = \rho(R)$. We tend to obtain the optimal synthesis of $AS_1,\ldots,AS_n$, based on a measure Ψ of classification algorithm quality.

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

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

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

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

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:

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

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

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

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

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

Proof. A similar result (for a problem formulated in a 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 an optimal subset of agents for a certain data table and a set of classifying agents. Let $\mathbf{B} = \{b_{ij}\}$ be an $n \times m$ binary matrix to be covered by a minimal set of columns (suppose that 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 matrix $\mathbf{B}$ corresponds to a pair of objects from U and every column of $\mathbf{B}$ corresponds to one attribute from A (hence $|A| = n$, $|U| = 2m$). Let attribute values be defined as follows:

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

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 subtable $\mathbb{A}_i$ and the subset of attributes $B_i = \{a_i\}$ (which is a reduct of $\mathbb{A}_i$):

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

The set U_i contains 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 (6). We will prove that classification algorithm CA_{AS_J} correctly classifies each object from U iff J corresponds to a 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 that 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,

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

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

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

It 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 **B**. Note that, by assumption (7), if there are many ensembles that classify every object in U, a function Ψ will prefer the smaller one. Hence, the optimal subset of agents corresponds to a minimal covering of **B**. This completes the construction of the (polynomial) transformation of the matrix covering problem to the problem of selecting an optimal subset of agents, which proves the NP-hardness of the latter. □

5 Adaptive Strategies of Constructing Classifiers

The 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 expert knowledge (understanding the application domain, the goals of the analytic process, selecting an appropriate data set, interpreting and using results). One of the important fields in KDD research is seeking to develop methods of possibly automating many steps of the KDD process by using, e.g., automatic feature extraction, data reduction, or algorithm selection via parameterization. These methods are often based on an 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 information granulation used at each level: feature extraction and preprocessing, data reduction and generalization, and synthesis of the final classifier (see Fig. 4). Some of these steps are known to be NP-hard, e.g., an optimal decomposition problem [11], optimal reduct finding (in the sense of its length or other measures, and also in cases 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 in Fig. 4 was presented by the author in [35]. On the lower level, feature extraction evolu-

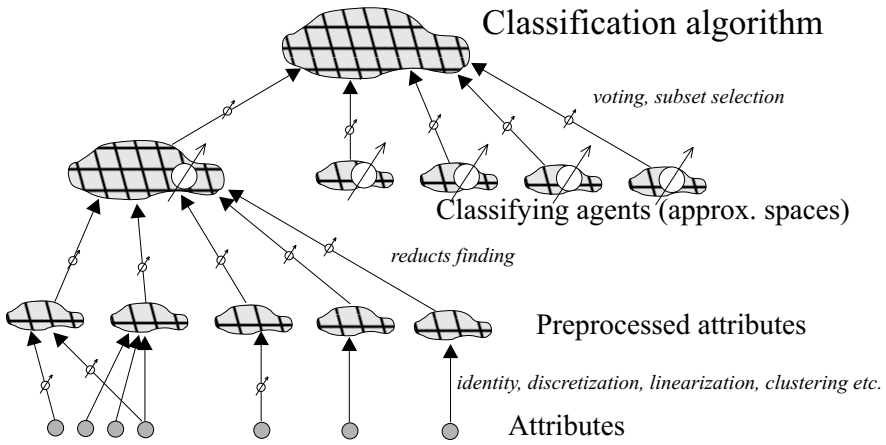

Fig. 4. Hierarchical construction of a classifying algorithm from granules (descriptors, approximation spaces); small circles with arrows denote adaptable parameters of information granules (or transforming/combining them)

tionary algorithms are used to create optimal linearization of attributes or new features based on a relational database (see Sect. 3). The process may be regarded as an optimization of weights in cases of linearization or as a selection (by 0–1 weights) of the best new attribute from the inductive closure of the database. There are other potential spaces of structures of new attributes, based on both a supervised and an unsupervised learning method. These spaces include clustering, PCA, discretization, and feature extraction methods used in cases 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 an algorithm. A group of adaptation-based evolutionary (hybrid) algorithms for reduct finding creates a complete approximation space (by providing a set of rules as a source of templates forming the $\mathcal{R}$ family) parameterized by approximation coefficients in cases 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 in Fig. 4 is concerned with creating optimal ensembles of classifying agents. The problem is NP-hard (see Theorem 1); the results of practical experiments confirm that increasing the number of agents in an ensemble does not necessarily lead to enhancing the classification results (see Fig. 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 the fitness function is calculated based on classification results of an additional testing subtable.

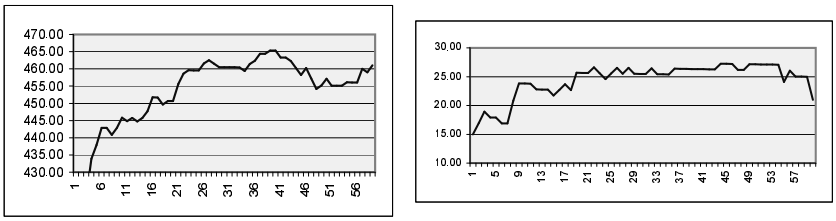

Fig. 5. Classification results (vertical axis) and number of agents in an ensemble (horizontal axis) – `DNA_splices` and `primary_tumor` data sets

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

One may notice an interesting analogy between Fig. 4 and neural networks [13], [18]. In a multilayer feedforward 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 back-propagation algorithm) is based on adjusting parameters (weights) based on the model prediction error, propagated down the net. There is no direct way to adapt this scheme to the general case of adaptive rule-based classifiers since there are 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

This chapter describes a general scheme of modeling a process of classification system construction using the notion of an information granule. The process starts

with a set of elementary information granules based on single attributes. The first level of the adaptive process of classifier construction is preprocessing of the initial attributes: discretization (generalizing several information granules into one), linearization (combining several attributes using an optimal, in some sense, linear combination of them where 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 a higher order). Rough set theory is a tool for generalizing descriptors (granules based on single attributes) onto the sets of rules.

The last level of the process described in the chapter is to combine a set of information granules (sets of rules, classifying agents) into one classification system and to resolve conflicts between them. The problem of selecting an optimal subset of agents 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 chapter. All the steps of granule combination are parameterized, and some algorithms for parameter optimization are presented. Quality measures based on (estimated) efficiency of classifying new cases are proposed.

Table 1. Experimental results compared with two popular classifiers. The result column contains a number (percent) of properly classified test objects

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

The adaptive classification system described above was partially implemented by the author [29, 34, 35]. The results of experiments on some benchmark data tables are presented in Table 1.

Further research is needed in many detailed aspects of the process described. Re-

gular examination of adaptive strategies of parameter optimization (especially when generalizing parameters, not only weights) should be performed. Although many parts of the process have been successfully implemented by the author, there are still no experimental results for 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.

Acknowledgments

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

References

1. Th. Bäck. An overview of parameter control methods by self-adaptation in evolutionary algorithms. *Fundamenta Informaticae*, 35(1): 51–66, 1998.
2. S.D. Bay. Combining nearest neighbor classifiers through multiple feature subsets. In *Proceedings of the 15th International Conference on Machine Learning (ICML'98)*, Morgan Kaufmann, San Mateo, CA, 1998.
3. J.G. Bazan, H.S. Nguyen, S.H. Nguyen, P. Synak, J. Wróblewski. Rough set algorithms in classification problem. In L. Polkowski, S. Tsumoto, and T.Y. Lin, editors, *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*, 49–88, Physica, Heidelberg, 2000.
4. I. Düntsch, G. Gediga. Uncertainty measures of rough set prediction. *Artificial Intelligence*, 106 : 77–107, 1998.
5. I. Düntsch, G. Gediga, H.S. Nguyen. Rough set data analysis in the KDD process. In *Proceedings of the 8th International Conference on Information Processing and Management under Uncertainty (IPMU 2000)*, 220–226, Madrid, Spain, 2000.
6. I.T. Jolliffe. *Principal Component Analysis*. Springer, Berlin, 1986.
7. M. Li, P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, New York, 1993.
8. T.Y. Lin, A.M. Wildberger, editors. *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery*. Simulation Councils, San Diego, CA, 1995.
9. H. Liu, H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer, Dordrecht, 1998.
10. S.H. Nguyen, L. Polkowski, A. Skowron, P. Synak, J. Wrblewski. Searching for approximate description of decision classes. In *Proceedings of the 4th International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, (RSFD'96)*, 153–161, Tokyo, 1996.
11. H.S. Nguyen. *Discretization of Real Value Attributes: Boolean Reasoning Approach*. Ph.D. Dissertation, Faculty of Mathematics, Informatics and Mechanics, Warsaw University, 2002.
12. H.S. Nguyen, A.Skowron, J.Stepaniuk. Granular computing: A rough set approach. *Computational Intelligence*, 17(3): 514–544, 2001.
13. S.K. Pal, W. Pedrycz, A. Skowron, R. Swiniarski, editors. Rough-neuro computing (special issue). Vol. 36 of *Neurocomputing: An International Journal*, 2001.

14. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.
15. L. Polkowski, A. Skowron, J. Zytkow. Tolerance based rough sets. In *[18]*, 55–58, 1994.
16. L. Polkowski, A. Skowron. Rough mereological foundations for design, analysis, synthesis and control in distributed systems. *Information Sciences*, 104(1/2): 129–156, 1998.
17. L. Polkowski, A. Skowron, editors. *Rough Sets in Knowledge Discovery* Vols. 1, 2. Physica, Heidelberg, 1998.
18. L. Polkowski, A. Skowron, editors. Rough-Neuro Computing. In *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC 2000)*, LNAI 2005, 25–32, Springer, Heidelberg, 2001.
19. RSES homepage – rough set based data analysis system. Available at http://loic.mimuw.edu.pl/˜rses/
20. A. Skowron. Approximation spaces in rough neurocomputing. In S. Hirano, M. Inuiguchi, S. Tsumoto, editors, *Rough Set Theory and Granular Computing*, Physica, Heidelberg, to appear.
21. A. Skowron, J. Stepaniuk. Approximation of relations. In *[38]*, 161–166, 1993.
22. A. Skowron, J. Stepaniuk. Generalized approximation spaces. In T.Y. Lin, A.M. Wildberger, editors, *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery* 18–21, Simulation Councils, San Diego, CA, 1995.
23. A. Skowron, J. Stepaniuk. Tolerance approximation spaces. *Fundamenta Informaticae*, 27: 245–253, 1996.
24. A. Skowron, J. Stepaniuk. Information granule decomposition. *Fundamenta Informaticae*, 47: 337–350, 2001.
25. R. Słowiński, D. Vanderpooten. *Similarity Relation as a Basis for Rough Approximations*. Report number 53/95 of the Institute of Computer Science, Warsaw University of Technology, 1995; see also P.P. Wang, editor, *Advances in Machine Intelligence & Soft Computing*, 17–33, Bookwrights, Raleigh, NC, 1997.
26. J. Stepaniuk. Approximation spaces, reducts and representatives. In L. Polkowski, A. Skowron, editors, *Rough Sets in Knowledge Discovery, Vol. 2*, 109–126, Physica, Heidelberg, 1998.
27. J. Stepaniuk. Knowledge discovery by application of rough set methods. In L. Polkowski, T.Y. Lin, S. Tsumoto, editors, *Rough Sets: New Developments in Knowledge Discovery in Information Systems*, Physica, Heidelberg, 2000.
28. D. Ślęzak. Approximate reducts in decision tables. In *Proceedings of the 7th International Conference on Information Processing and Management under Uncertainty (IPMU'96)*, 1159–1164, Universidad da Granada, Granada, 1996.
29. D. Ślęzak, J.Wróblewski. Classification algorithms based on linear combinations of features. In *Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'99)*, LNAI 1704, 548–553, Springer, Berlin, 1999.
30. D. Ślęzak, J. Wróblewski. Application of normalized decision measures to the new case classification. In *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC 2000)*, LNAI 2005, 515–522, Springer, Berlin, 2000.
31. D. Ślęzak. *Approximate Decision Reducts*. Ph.D. Dissertation, Faculty of Mathematics, Informatics and Mechanics, Warsaw University, 2002 (in Polish).
32. P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. MIT Press, Cambridge, MA, 2000.

33. J. Wróblewski. Analyzing relational databases using rough set based methods. In *Proceedings of the 8th Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2000)*, 256–262, Madrid, Spain, 2000.
34. J. Wróblewski. Ensembles of classifiers based on approximate reducts. In *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P 2000)*, volume 140(2) of *Informatik-Bericht*, 355–362, Humboldt-Universität, Berlin, 2000; also in *Fundamenta Informaticae*, 47(3/4): 351–360, 2001.
35. J. Wróblewski. *Adaptive Methods of Object Classification*. Ph.D. Dissertation, Faculty of Mathematics, Informatics and Mechanics, Warsaw University, 2002 (in Polish).
36. L.A. Zadeh, J.Kacprzyk, editors. *Computing with Words in Information/Intelligent Systems*, Vols. 1, 2. Physica, Heidelberg, 1999.
37. W. Ziarko. Variable precision rough set model. *Journal of Computer and System Sciences*, 46: 39–59, 1993.
38. W. Ziarko, editor. *Proceedings of the International Workshop on Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93)*, Workshops in Computing, Springer & British Computer Society, London, Berlin, 1994.
39. W. Ziarko. Approximation region-based decision tables. In *Proceedings of the 1st International Conference on Rough Sets and Current Trends in Computing (RSCTC'98)*, LNAI 1424, 178–185, Springer, Berlin, 1998.