Chapter 25
Information Granulation and Pattern Recognition

Andrzej Skowron,[1] Roman W. Swiniarski[2]

[1] Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[2] San Diego State University, Department of Mathematical and Computer Sciences, 5500 Campanile Drive, San Diego, CA 92182, USA
rswiniar@sciences.sdsu.edu

Summary. We discuss information granulation applications in pattern recognition. The chapter consists of two parts. In the first part, we present applications of rough set methods for feature selection in pattern recognition. We emphasize the role of different forms of reducts that are the basic constructs of the rough set approach in feature selection. In the overview of methods for feature selection, we discuss feature selection criteria based on the rough set approach and the relationships between them and other existing criteria. Our algorithm for feature selection used in the application reported is based on an application of the rough set method to the result of principal component analysis used for feature projection and reduction. Finally, the first part presents numerical results of face recognition experiments using a neural network, with feature selection based on proposed principal component analysis and rough set methods. The second part consists of an outline of an approach to pattern recognition with the application of background knowledge specified in natural language. The approach is based on constructing approximations of reasoning schemes. Such approximations are called approximate reasoning schemes and rough neural networks.

1 Introduction

Reduction of pattern dimensionality via feature extraction and feature selection [9,17,21,22] is among the most fundamentals steps in data preprocessing. We present rough sets methods and principal components analysis (PCA) in the context of feature selection in pattern recognition.

The chapter begins with a short introduction to rough set theory [28]. We emphasize the special role of reducts in feature selection, including dynamic reducts [2,4,5]. Then, we present a short overview of a feature selection problem including open-loop and closed-loop feature selection methods [9]. This section focuses the discussion on feature selection criteria, including rough set based methods. The next section presents a short description of principal component analysis [9] as a method of feature projection and reduction. It also contains a description of rough set-based methods, proposed jointly with principal component analysis, for feature projection and reduction. The following section describes the results of numerical experiments

of face recognition using rough set based methods for feature selection and neural networks. This section also contains a short description of feature extraction from facial images using singular value decomposition (SVD).

The second part of the chapter consists of an outline of an approach, called the rough-neurocomputing approach (see Chaps. 2 and 3), for pattern recognition with an application of background knowledge specified in natural language. The approach is based on a rough mereological approach (see, e.g., [35]) for information granule calculi. The goal of information granule calculi is to make it possible to imitate reasoning in natural language by means of information granules. Such granules have a complex information structure representing approximations of reasoning schemes in natural language over vague concepts. Reasoning schemes in natural language are built over vague concepts and relations between them creating ontologies. In natural language we call them approximation reasoning schemes (AR schemes) or rough neural networks. Our approach to pattern recognition is based on searching for clusters of objects close to a given standard (prototype) to a given degree. Using the rough set approach, one can interpret such standards as the lower approximations of concepts. Moreover, methods for extracting special relationships, called productions, between such clusters are emphasized. They correspond to local relationships between concepts from background knowledge. They make it possible to conclude that a target concept is satisfied to a satisfactory degree for a given object if the input concepts are satisfied to some satisfactory degree by input patterns related to the object. A special method for composing such productions leads to derivations of robust AR schemes. Any AR scheme guarantees that the target concept of such a scheme is satisfied to a satisfactory degree for a given object if the input concepts for this scheme are satisfied to some satisfactory degree by the object. AR schemes are then used to induce approximations of more complex concepts from a knowledge base, assuming that classifiers representing some primitive concepts have been constructed. In the second part of the chapter, we outline the approach, and we present an illustrative example.

2 Preliminaries of Rough Sets

Rough set theory was introduced by Zdzisław Pawlak (see, e.g., [18,28]) to deal with imprecise or vague concepts. In recent years, we have witnessed a rapid growth of interest in rough set theory and its applications worldwide (see, e.g., [18, 30, 33, 34, 45, 52].

In this section, we present the basic concepts of rough set theory and some of its extensions. A variety of methods for generating decision rules, reduct computation, and continuous variable discretization are very important issues not discussed here. We emphasize only the developed methodology based on discernibility and Boolean reasoning for efficient computation of different constructs, including reducts and decision rules.

Many other important issues are not covered here. Let us mention some of them. The relationship of rough set theory to many other theories has been extensively investigated. In particular, its relationships to fuzzy set theory, the theory of evidence, Boolean reasoning methods, statistical methods, and decision theory have been clarified and seem to be thoroughly understood. There are reports on many hybrid methods obtained by combining the rough set approach with others, such as fuzzy sets, neural networks, genetic algorithms, principal component analysis, and singular value decomposition [27]. Recently, it has been shown that the rough set approach can be used for synthesizing concept approximations in a distributed environment of intelligent agents. These issues related to various logics related to rough sets and many advanced algebraic properties of rough sets are also not covered here. Readers interested in these issues are advised to consult [18,32,33,44] and the bibliography included in these books and articles.

2.1 Basic Approach

The rough set approach is founded on the assumption that with every object of a universe of discourse, we associate some information (data, knowledge). For example, if objects are patients suffering from a certain disease, then the symptoms of the disease form information about patients. Objects characterized by the same information are indiscernible (similar) in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory.

Any set of all indiscernible (similar) objects is called an elementary set and forms a basic granule (atom) of knowledge about a universe. Any union of some elementary sets is referred to as crisp (precise) set — otherwise, the set is rough (imprecise, vague).

Consequently, each rough set has boundary-line cases, i.e., objects that cannot be classified with certainty either as members of the set or of its complement. Obviously, crisp sets have no boundary-line elements at all. That means that boundary-line cases cannot be properly classified by employing the available knowledge.

Thus, the assumption that objects can be "seen" only through the information available about them leads to the view that knowledge has a granular structure. Due to the granularity of knowledge, some objects of interest cannot be discerned and appear the same (or similar). As a consequence, vague concepts (in contrast to precise or crisp concepts) cannot be characterized in terms of information about their elements. Therefore, in the proposed approach, we assume that any vague concept is replaced by a pair of precise concepts — called the lower and the upper approximations of the vague concept. The lower approximation consists of all objects that surely belong to the concept, and the upper approximation contains all objects that possibly belong

to the concept. Obviously, the difference between the upper and the lower approximations constitutes the boundary region of the vague concept. Approximations are two basic operations in rough set theory.

2.2 Approximations and Rough Sets

We have mentioned in Sect. 2.1 that the starting point of rough set theory is the indiscernibility relation, generated by information about objects of interest. The indiscernibility relation is intended to express the fact that due to the lack of knowledge, we are unable to discern some objects by employing the available information. It means that, in general, we are unable to deal with each particular object, but we have to consider clusters of indiscernible objects as fundamental concepts of our theory.

Suppose that we are given two finite, nonempty sets U and A, where U is the *universe* of *objects, cases*, and A is a set of *attributes, features*. The pair $IS = (U,A)$ is called an *information table*. With every attribute $a \in A$, we associate a set V_a, of its *values*, called the *domain* of a. By $\mathbf{a}(x)$ we denote a data pattern $(a_1(x),\ldots,a_n(x))$ defined by the object x and attributes from $A = \{a_1,\ldots,a_n\}$. A data pattern of IS is any feature value vector $\mathbf{v} = (v_1,\ldots,v_n)$ where $v_i \in V_{a_i}$ for $i = 1,\ldots,n$ such that $\mathbf{v} = \mathbf{a}(x)$ for some $x \in U$.

Any subset B of A determines a binary relation $I(B)$ on U, called the *indiscernibility relation*, defined by

$$xI(B)y \text{ if and only if } a(x) = a(y) \text{ for every } a \in B, \tag{1}$$

where $a(x)$ denotes the value of attribute a for object x.

Obviously $I(B)$ is an equivalence relation. The family of all equivalence classes of $I(B)$, i.e., the partition determined by B, will be denoted by $U/I(B)$, or simply U/B; an equivalence class of $I(B)$, i.e., the block of the partition U/B containing x, will be denoted by $B(x)$.

If $(x,y) \in I(B)$, we will say that x and y are *B-indiscernible*. Equivalence classes of the relation $I(B)$ (or blocks of the partition U/B) are referred to as *B-elementary sets*. In the rough set approach, elementary sets are the basic building blocks (concepts) of our knowledge about reality. The unions of *B-elementary sets* are called *B-definable sets.*

The indiscernibility relation will be further used to define basic concepts of rough set theory. Let us define now the following two operations on sets:

$$B_*(X) = \{x \in U : B(x) \subseteq X\}, \tag{2}$$

$$B^*(X) = \{x \in U : B(x) \cap X \neq \emptyset\}, \tag{3}$$

assigning to every subset X of the universe U two sets $B_*(X)$ and $B^*(X)$ called the *B-lower* and the *B-upper approximation* of X, respectively. The set,

$$BN_B(X) = B^*(X) - B_*(X), \tag{4}$$

will be referred to as the *B-boundary region* of X.

If the boundary region of X is the empty set, i.e., $BN_B(X) = \emptyset$, then the set X is *crisp* (*exact*) with respect to B; in the opposite case, i.e., if $BN_B(X) \neq \emptyset$, the set X is referred to as *rough* (*inexact*) with respect to B.

A rough set can be also characterized numerically, e.g., by the following coefficient:

$$\alpha_B(X) = \frac{|B_*(X)|}{|B^*(X)|}, \tag{5}$$

called the *accuracy of approximation*, where $|X|$ denotes the cardinality of $X \neq \emptyset$. Obviously, $0 \leq \alpha_B(X) \leq 1$. If $\alpha_B(X) = 1$, then X is *crisp* with respect to B (X is *precise* with respect to B), and otherwise, if $\alpha_B(X) < 1$, then X is *rough* with respect to B (X is *vague* with respect to B).

Several generalizations of the rough set approach based on approximation spaces defined by (U,R), where R is an equivalence relation (called the indiscernibility relation) in U, have been reported in the literature (for references, see the papers and bibliography in [18,32,33,44]. Let us mention two of them.

A generalized approximation space can be defined as $AS = (U,I,\nu)$ where I is the *uncertainty function* defined on U with values in the power set $P(U)$ of U [$I(x)$ is the *neighborhood* of x] and ν is the *inclusion function* defined on the Cartesian product $P(U) \times P(U)$ with values in the interval $[0,1]$ measuring the degree of inclusion of sets. The lower AS_* and upper AS^* approximation operations can be defined in AS by

$$AS_*(X) = \{x \in U : \nu(I(x),X) = 1\}, \tag{6}$$

$$AS^*(X) = \{x \in U : \nu(I(x),X) > 0\}. \tag{7}$$

In the case discussed above, $I(x)$ is equal to the equivalence class $B(x)$ of the indiscernibility relation $I(B)$; when a tolerance (similarity) relation $\tau \subseteq U \times U$ is given, we let $I(x) = \{y \in U : x\tau y\}$, i.e., $I(x)$ is equal to the tolerance class of τ defined by x. The standard inclusion relation is defined by $\nu(X,Y) = \frac{|X \cap Y|}{|X|}$ if X is nonempty, and otherwise, $\nu(X,Y) = 1$. For applications, it is important to have some constructive definitions of I and ν.

One can consider another way to define $I(x)$. Usually, together with AS, we consider some set F of formulas describing sets of objects in the universe U of AS

defined by semantics $\|\cdot\|_{AS}$, i.e., $\|\alpha\|_{AS} \subseteq U$ for any $\alpha \in F$. Now, one can take the set,

$$N_F(x) = \{\alpha \in F : x \in \|\alpha\|_{AS}\} \text{ and } I(x) = \|\alpha\|_{AS}, \tag{8}$$

where α is selected or constructed from $N_F(x)$. Hence, more general uncertainty functions having values in $P[P(U)]$ can be defined (see also Chap. 3). The parametric approximation spaces are examples of such approximation spaces. These spaces have interesting applications. For example, by tuning their parameters, one can search for the optimal, under chosen criteria (e.g., the minimal description length), approximation space for a concept description.

The approach based on inclusion functions has been generalized to the *rough mereological approach*. The *inclusion relation* $x\mu_r y$ with intended meaning *x is part of y to a degree r* has been taken as the basic notion of *rough mereology* that is a generalization of Leśniewski mereology. Rough mereology offers a methodology for synthesizing and analyzing objects in a distributed environment of intelligent agents, in particular, for synthesizing of objects satisfying a given specification in satisfactory degree or for control in such complex environment. Moreover, rough mereology has been recently used for developing foundations of *information granule calculus*, an attempt toward formalization of the computing with words paradigm recently formulated by Lotfi Zadeh [58]. Research on rough mereology has shown the importance of another notion, namely, the *closeness* of complex objects (e.g., concepts). This can be defined by $xcl_{r,r'}y$ if and only if $x\mu_r y$ and $y\mu_{r'}x$. The inclusion and closeness definitions of complex information granules are dependent on applications. However, it is possible to define the granule syntax and semantics as a basis for the inclusion and closeness definitions.

Finally, let us mention that approximation spaces are usually defined as parameterized approximation spaces. In the simplest case, the parameter set is defined by the power set of a given feature set. By parameter tuning, the relevant approximation space is selected for a given data set and target task.

2.3 Rough Sets and Membership Function

Rough sets can also be introduced by using a *rough membership function*, defined by

$$\mu_X^B(x) = \frac{|X \cap B(x)|}{|B(x)|}. \tag{9}$$

Obviously, $0 \leq \mu_X^B(x) \leq 1$. Hence, the value of the membership function for a given object x can be interpreted as the degree of overlap between the indiscernibility class of x and the set X. One can also interpret this value as the conditional probability that an object from the indiscernibility class defined by x belongs to X.

The rough membership function can be used to define approximations and the boundary region of a set, as shown here:

$$B_*(X) = \{x \in U : \mu_X^B(x) = 1\}, \tag{10}$$

$$B^*(X) = \{x \in U : \mu_X^B(x) > 0\}, \tag{11}$$

$$BN_B(X) = \{x \in U : 0 < \mu_X^B(x) < 1\}. \tag{12}$$

2.4 Decision Tables and Decision Rules

Sometimes, in an information table (U,A), it is useful to distinguish a partition of A into two classes $C,D \subseteq A$ of attributes, called *condition* and *decision* (*action*) attributes, respectively. The tuple $DT = (U,C,D)$ is called a *decision table (system)*. Any such decision table where $U = \{u_1,\ldots,u_N\}$, $C = \{a_1,\ldots,a_n\}$ and $D = \{d_1,\ldots,d_k\}$ can be represented by a data sequence (also called data set) of data patterns $((\mathbf{v}_1,\mathbf{target}_1),\ldots,(\mathbf{v}_N,\mathbf{target}_N))$, where $\mathbf{v}_i = \mathbf{C}(x_i)$, $\mathbf{target}_i = \mathbf{D}(x_i)$, and $\mathbf{C}_i = (a_1(x_i),\ldots,a_n(x_i))$, $\mathbf{D}_i = (d_1(x_i),\ldots,d_k(x_i))$, for $i = 1,\ldots,N$. It is obvious that any data sequence also defines a decision table. The equivalence classes of $I(D)$ are called decision classes.

Let $V = \bigcup\{V_a \mid a \in C\} \cup V_d$. Atomic formulas over $B \subseteq C \cup D$ and V are expressions $a = v$ called *descriptors* (*selectors*) over B and V, where $a \in B$ and $v \in V_a$. The set $\mathcal{F}(B,V)$ of formulas over B and V is the least set containing all atomic formulas over B and V and closed with respect to the propositional connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$ (negation).

By $\|\varphi\|_{DT}$, we denote the meaning of $\varphi \in \mathcal{F}(B,V)$ in the decision table DT which is the set of all objects in U with the property φ. These sets are defined as follows: $\|a = v\|_{DT} = \{x \in U \mid a(x) = v\}$, $\|\varphi \wedge \varphi'\|_{DT} = \|\varphi\|_{DT} \cap \|\varphi'\|_{DT}$; $\|\varphi \vee \varphi'\|_{DT} = \|\varphi\|_{DT} \cup \|\varphi'\|_{DT}$; $\|\neg\varphi\|_{DT} = U - \|\varphi\|_{DT}$.

The formulas from $\mathcal{F}(C,V)$, $\mathcal{F}(D,V)$ are called *condition formulas of* DT and *decision formulas of* DT, respectively.

Any object $x \in U$ belongs to a *decision class* $\|\bigwedge_{a\in D} a = a(x)\|_{DT}$ of DT. All decision classes of DT create a partition of the universe U.

A *decision rule* for DT is any expression of the form $\varphi \Rightarrow \psi$, where $\varphi \in \mathcal{F}(C,V)$, $\psi \in \mathcal{F}(D,V)$, and $\|\varphi\|_{DT} \neq \emptyset$. Formulas φ and ψ are referred to as the *predecessor* and the *successor* of decision rule $\varphi \Rightarrow \psi$. Decision rules are often called "$IF \ldots THEN \ldots$" rules.

Decision rule $\varphi \Rightarrow \psi$ is *true* in, DT if and only if $\|\varphi\|_{DT} \subseteq \|\psi\|_{DT}$. Otherwise, one can measure its *truth degree* by introducing some inclusion measure of $\|\varphi\|_{DT}$ in $\|\psi\|_{DT}$ (see Chap. 3).

Each object x of a decision table determines a *decision rule*,

$$\bigwedge_{a\in C} a = a(x) \Rightarrow \bigwedge_{a\in D} a = a(x). \tag{13}$$

Decision rules corresponding to some objects can have the same condition parts but different decision parts. Such rules are called *inconsistent* (*nondeterministic, conflicting, possible*); otherwise, the rules are referred to as *consistent* (*certain, sure, deterministic, nonconflicting*) rules. Decision tables containing inconsistent decision rules are called *inconsistent* (*nondeterministic, conflicting*); otherwise, the table is *consistent* (*deterministic, nonconflicting*).

When a set of rules has been induced from a decision table containing a set of training examples, they can be inspected to see if they reveal any novel relationships between attributes that are worth pursuing for further research. Furthermore, the rules can be applied to a set of unseen cases to estimate their classificatory power. For a systematic overview of rule application methods, the reader is referred to bibliographies included in [18,32,33,44].

2.5 Dependency of Attributes

Another important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes D depends totally on a set of attributes C, denoted $C \Rightarrow D$, if the values of attributes from C uniquely determine the values of attributes from D. In other words, D depends totally on C, if there exists a functional dependency between values of C and D.

Formally, dependency can be defined in the following way. Let D and C be subsets of A.

We will say that D *depends on* C to a *degree* k $(0 \leq k \leq 1)$, denoted $C \Rightarrow_k D$, if

$$k = \gamma(C,D) = \frac{|POS_C(D)|}{|U|}, \tag{14}$$

where

$$POS_C(D) = \bigcup_{X\in U/D} C_*(X), \tag{15}$$

called a *positive region* of the partition U/D with respect to C, is the set of all elements of U that can be uniquely classified in blocks of the partition U/D by means of C. If $k = 1$, we say that D *depends totally* on C, and if $k < 1$, we say that D *depends partially* (to a *degree* k) on C. The coefficient k expresses the ratio of all elements of the universe, which can be properly classified in blocks of the partition U/D, employing attributes C and will be called the *degree of the dependency.* It can

be easily seen that if D depends totally on C, then $I(C) \subseteq I(D)$. This means that the partition generated by C is finer than the partition generated by D. Notice that the concept of dependency discussed above corresponds to that considered in relational databases. Summing up D, is *totally* (*partially*) dependent on C, if *all* (*some*) elements of the universe U can be uniquely classified in blocks of the partition U/D, employing C. The coefficient $1-\gamma(C,D)$ can be called the inconsistency degree of the DT [24].

2.6 Discernibility and Boolean Reasoning

The ability to discern between perceived objects is important in constructing many entities such as reducts, decision rules, and decision algorithms. In the classical rough set approach, the *discernibility relation* $DIS(B) \subseteq U \times U$ is defined by

$$xDIS(B)y \text{ if and only if } non\,[xI(B)y]. \tag{16}$$

However, this is generally not the case for generalized approximation spaces [one can define indiscernibility by $x \in I(y)$ and discernibility by $I(x) \cap I(y) = \emptyset$ for any objects x, y].

Boolean reasoning [7,8,42] is based on constructing for a given problem P, a corresponding Boolean function f_P with the following property: the solutions of problem P can be decoded from prime implicants of the Boolean function f_P. Let us mention that to solve real-life problems, it is necessary to deal with Boolean functions that have a huge size and a large number of variables.

A successful methodology based on the discernibility of objects and Boolean reasoning has been developed for computing many important, for applications entities such as reducts and their approximations (see the following section), decision rules, association rules, discretization of real value attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher order surfaces, pattern extraction from data as well as conflict resolution or negotiation (for references, see the papers and bibliographies in [18,32,33,44]).

Most of the problems related to generating of the above mentioned entities are NP-complete or NP-hard [46]. However, it was possible to develop efficient heuristics returning suboptimal solutions of the problems. The results of experiments on many data sets are very promising. They show very good quality of solutions generated by the heuristics in comparison with other methods reported in the literature (e.g., with respect to the classification quality of unseen objects). Moreover, they are very efficient from the point of view of time necessary for computing the solution.

It is important to note that the methodology allows us to construct heuristics having

a very important *approximation property* which can be formulated as follows: expressions generated by heuristics (i.e., implicants) *close* to prime implicants define approximate solutions for the problem.

2.7 Reduction of Attributes

We often face the question whether we can remove some data from a data table and preserve its basic properties, that is, whether a table contains some superfluous data. Let us express this idea more precisely.

Given an information system IS, a *reduct* is a minimal set of attributes $B \subseteq A$ such that $I(A) = I(B)$. In other words, a reduct is a minimal set of attributes from A that preserves the original classification defined by the set A of attributes. Finding a minimal reduct is NP-hard; one can also show that for any m (sufficiently large), there exists an information system with m attributes having a number of reducts exponential in m. There exist, fortunately, good heuristics that compute sufficiently many reducts with the required properties (e.g., related to their length) in an acceptable time.

Let IS be an information system with n objects. The *discernibility matrix* of IS is a symmetrical $n \times n$ matrix with entries c_{ij} as given below. Each entry consists of the set of attributes upon which objects x_i and x_j differ.

$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \quad \text{for} \quad i, j = 1, \ldots, n. \tag{17}$$

A *discernibility function* f_{IS} for an information system IS is a Boolean function of m Boolean variables $a_1^*, \ldots, a_m^*$ (corresponding to the attributes $a_1, \ldots, a_m$) defined by

$$f_{IS}(a_1^*, \ldots, a_m^*) = \bigwedge \left\{ \bigvee c_{ij}^* \mid 1 \leq j \leq i \leq n,\, c_{ij} \neq \emptyset \right\}, \tag{18}$$

where $c_{ij}^* = \{a^* \mid a \in c_{ij}\}$. In the sequel, we will write a_i instead of a_i^*.

The discernibility function f_{IS} describes constraints which should be preserved if one would like to preserve discernibility between all pairs of discernible objects from IS. It requires us to keep at least one attribute from each nonempty entry of the discernibility matrix, i.e., corresponding to any pair of discernible objects. One can show [46] that the sets of all minimal sets of attributes preserving discernibility between objects, i.e., reducts correspond to prime implicants of the discernibility function f_{IS}.

The intersection of all reducts is the so-called *core*. It is well known that choosing a random reduct as a relevant set of features in an information system will give rather poor results. Hence, several techniques have been developed to select relevant reducts or their approximations. Among them is one based on so-called *dynamic reducts* [2,4]. The attributes are considered relevant if they belong to dynamic

reducts with a sufficiently high stability coefficient, i.e., they appear with sufficiently high frequency in random samples extracted from a given information system.

There are several kinds of reducts considered for decision tables. We will discuss one of them. Let $\mathcal{A} = (U, A, d)$ be a decision system (i.e., we assume, for simplicity of notation that the set D of decision attributes consists of only one element d, $D = \{d\}$ and $C = A$). The *generalized decision in* $\mathcal{A}$ is the function $\partial_A : U \longrightarrow \mathcal{P}(V_d)$ defined by

$$\partial_A(x) = \{i \mid \exists x' \in U \; x' \, IND(A) \, x \text{ and } d(x') = i\}. \tag{19}$$

A decision system $\mathcal{A}$ is called *consistent (deterministic)*, if $|\partial_A(x)| = 1$ for any $x \in U$, otherwise, $\mathcal{A}$ is *inconsistent (nondeterministic)*. Any set consisting of all objects with the same generalized decision value is called a *generalized decision class*. Decision classes are denoted by C_i, where the subscript denotes the decision value.

It is easy to see that a decision system $\mathcal{A}$ is consistent if and only if $POS_A(d) = U$. Moreover, if $\partial_B = \partial_{B'}$, then $POS_B(d) = POS_{B'}(d)$ for any pair of nonempty sets $B, B' \subseteq A$. Hence, the definition of a decision-relative reduct: a subset $B \subseteq A$ is a *relative reduct* if it is a minimal set such that $POS_A(d) = POS_B(d)$. Decision-relative reducts may be found from a discernibility matrix $M^d(\mathcal{A}) = (c_{ij}^d)$ assuming

$$c_{ij}^d = \begin{cases} c_{ij} - \{d\} \text{ if } (|\partial_A(x_i)| = 1 \text{ or } |\partial_A(x_j)| = 1) \\ \emptyset \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{20}$$

Matrix $M^d(\mathcal{A})$ is called *the decision-relative discernibility matrix of* $\mathcal{A}$. Construction of *the decision-relative discernibility function* from this matrix follows the construction of the discernibility function from the discernibility matrix. One can show that the set of *prime implicants* of $f_M^d(\mathcal{A})$ defines the set of all *decision-relative reducts* of $\mathcal{A}$.

Since the core is the intersection of all reducts, it is included in every reduct, i.e., each element of the core belongs to some reduct. Thus, the core is the most important subset of attributes since none of its elements can be removed without affecting the classification power of attributes.

Yet another kind of reduct, called reduct relative to objects, can be used for generating minimal decision rules from decision tables ([18,44]).

In some applications, instead of reducts, we prefer to use their approximations called α-reducts, where $\alpha \in [0, 1]$ is a real parameter. For a given information system $\mathcal{A} = (U, A)$, the set of attributes $B \subseteq A$ is called α-reduct if B has a nonempty intersection with at least $\alpha \cdot 100\%$ of nonempty sets $c_{i,j}$ of the discernibility matrix of $\mathcal{A}$.

Different kinds of reducts and their approximations are discussed in the literature as basic constructs for reasoning about data represented in information systems or

decision tables (see, e.g., [3,48,49]). It turns out that they can be efficiently computed using heuristics based on the Boolean reasoning approach.

3 Feature Selection

Feature selection is a process of finding a subset of features from the original set of features forming patterns in a given data set, optimal according to the given goal of processing and the criterion. An optimal feature selection is a process of finding a subset,

$$A_{opt} = \{a_{1,opt}, a_{2,opt}, \ldots, a_{m,opt}\}, \tag{21}$$

of A, which guarantees accomplishing a processing goal by minimizing a defined feature selection criterion $J_{\text{feature}}(A_{\text{feature\_subset}})$. A solution of an optimal feature selection does not need to be unique.

One can distinguish two paradigms in data model building and potentially, in an optimal feature selection (*minimum construction paradigms*): *the Occam's razor* and *minimum description length principle* [40].

By virtue of the minimum construction idea, one of the techniques for best feature selection could be based on choosing a minimal feature subset that fully describes all concepts (for example, classes in prediction-classification) in a given data set [1,28]. Let us call this paradigm *a minimum concept description*. However, this approach, good for a given (possibly limited) data set, may not be appropriate for processing unseen patterns. A robust processing algorithm with an associated set of features (reflecting complexity) is a trade-off between the ability to process a given data set versus generalization ability.

The second general paradigm of optimal feature selection, mainly used in classifier design, relates to selecting a feature subset that guarantees the maximal between-class separability for reduced data sets. This relates to the discriminatory power of features.

Feature selection methods consists of two main streams [6,11,13,16]: *open-loop methods* and *closed-loop methods*.

Open loop methods (*filter method*) are based mostly on selecting features using a between-class separability criterion [9,11]. They do not use feedback from predictor quality for the feature selection process.

Closed-loop methods [16] also called *wrapper methods*, are based on feature selection using *predictor (classifier) performance* (and thus forming feedback in processing) as a criterion of feature subset selection. A selected feature subset is evaluated using as a criterion, $J_{\text{feature}} = J_{\text{predictor}}$ a performance evaluation $J_{\text{predictor}}$ of

a whole prediction algorithm for the reduced data set containing patterns with the selected features as patterns elements.

Let us consider the problem of defining a feature selection criterion for a prediction task based on an original data set T containing N cases $(\mathbf{a}, \mathbf{target})$ constituted of *n-dimensional* input patterns **a** and a **target** pattern of output. Assume that the *m*-feature subset $A_{\text{feature}} \subseteq A$ ought to be evaluated on the basis of the closed-loop type criterion. A reduced data set T_{feature}, with patterns containing only *m*-features from the subset A_{feature}, should be constructed. Then, a type of predictor PR_{feature} (for example, *k*-nearest neighbors, or neural network), used for feature quality evaluation, should be decided. This predictor ideally should be the same as a final predictor PR for a whole design; however, in a simplified suboptimal solution, a computationally less expensive predictor can be used only for feature selection. Let us assume that, for the feature set A considered, a reduced feature data set A_{feature} has been selected and a predictor algorithm PR_{feature} based on A_{feature}, used for feature evaluation, decided. Then, evaluation of feature quality can be provided by using one of the methods used for the final predictor evaluation. This will require defining a performance criterion, $J_{PR_{\text{feature}}}$, of a predictor PR_{feature}, and an error counting method that will show how to estimate performance by averaging results. Consider as an example, a holdout error counting method for predictor performance evaluation. To evaluate the performance of a predictor PR_{feature}, an extracted feature data set T_{feature} is split into a N_{tra} case training set $T_{\text{feature,tra}}$ and a N_{test} case test set $T_{\text{feature,test}}$ (holdout for testing). Each case $(\mathbf{a}_f^i, \mathbf{target}^i)$ of both sets contains a feature pattern $\mathbf{a}_f^i$ labeled by $\mathbf{target}^i$. The evaluation criteria can be defined separately for prediction classification and prediction regression.

We will consider a defining feature selection criterion for a prediction classification task, when a feature subset T_{feature} case contains pairs $(\mathbf{a}_f, c_{\text{target}})$ of a feature input pattern $\mathbf{a}_f$ and a categorical type target c_{target} taking a value corresponding to one of the possible r decision classes C_i. The quality of classifier PR_{feature}, computed on the basis of the limited size test set $T_{\text{feature,test}}$ with N_{test} patterns, can be measured by using the following performance criterion $J_{PR_{\text{feature}}}$ (here equal to a feature selection criterion J_{feature}):

$$J_{PR_{\text{feature}}} = \hat{J}_{\text{all miscl}} = \frac{n_{\text{all miscl}}}{N_{\text{test}}} \cdot 100\%, \tag{22}$$

where $n_{\text{all miscl}}$ is the number of all misclassified patterns and N_{test} is the number of all tested patterns. This criterion estimates the probability of error from the relative frequency of error. Usually, cross-validation techniques are used to obtain better estimation of predictor quality.

An overview of feature selection methods can be found in [22,23]. Let us only mention that several methods of feature selection are inherently built into a predictor design procedure [39] and some methods of feature selection merge feature extraction with feature selection. A feature reduction (pruning) method for a self-

organizing neural network map, based on concept description, is suggested in [25].

We will concentrate in this chapter on the rough set approach to feature selection and on some relationships of rough set methods with existing ones.

3.1 Feature Selection Based on Rough Sets

The rough set approach to feature selection can be based on the minimal description length principle [40] and methods for tuning parameters of approximation spaces to obtain high-quality classifiers based on selected features. We have mentioned before an example of such parameter with possible values in the power set of the feature set, i.e., related to feature selection. Other parameters can be used, e.g., to measure the closeness of concepts [44].

One can distinguish two main steps in this approach.

In the first step, by using Boolean reasoning, relevant kinds of reducts from given data tables are extracted. These reducts preserve exactly the discernibility (and some other) constraints (e.g., reducts relative to objects for minimal decision rule generation).

In the second step, reduct approximations are extracted by parameter tuning. These reduct approximations allow shorter concept description than the exact reducts, and they still preserve the constraints to a sufficient degree to guarantee, e.g., sufficient approximation quality of the described (induced) concept [44].

In using rough sets for feature selection, two cases can be distinguished, global and local feature selection schemes. In the former case, the relevant attributes for the whole data table are selected, whereas in the latter case the descriptors of the form, (a, v) where $a \in A$ and $v \in V_a$, are selected for a given object. In both cases, we are searching for relevant features for object classification. In the global case, we are searching for features defining a partition (or covering) of the object universe. This partition should be relevant for describing the approximation of a partition (or part of it) defined by decision attribute. In the local case, we are extracting descriptors defining a relevant neighborhood for a given object with respect to a decision class.

Using rough sets [2,4,28,52] for feature selection was proposed in several contributions (see, e.g., [53,54]). The simplest approach is based on calculation of a core for a discrete attribute data set containing strongly relevant features and reducts containing a core plus additional weakly relevant features, such that each reduct is satisfactory for description of concepts in the data set. Based on a set of reducts for a data set, some criteria for feature selection can be formed, for example, selecting features from a minimal reduct, i.e., a reduct containing a minimal set of

attributes. Dynamic reducts were proposed to find a robust (well-generalizing) feature subset [2,4]. The selection of a dynamic reduct is based on the cross-validation method. Methods of dynamic reduct generation have been applied to relevant feature extraction, e.g., for dynamic selection of features represented in discretization as well as in the process of inducing relevant decision rules. Some other methods based on noninvasive data analysis and rough sets are reported in [12]. Let us now summarize the applications of rough set methods for feature selection in a closed loop. The method is based on searching first for short (dynamic) reducts or reduct approximations. This step can be realized using, for example, software systems such as ROSETTA (see http://www.idi.ntnu.no/~ aleks/rosetta/rosetta.html) or RSES (see alfa.mimuw.edu.pl). It can be based on genetic algorithms with the fitness function measuring the quality of the selected reduct approximation B-dependent, among others, on (1) the quality of the reduct approximation by the set B; (2) the cardinality of the feature set B; (3) the discernibility power of the feature set B with respect to the discernibility between decision classes measured, e.g., by means of the approximation quality of a D-reduct by B; (4) the number of equivalence classes created by a feature set on a given data set and/or the number of rules generated by this set [57]; (5) the closeness of concepts [44]; and (6) the conflict resolution strategy [55]. The parameters used to specify and compose the above components into a fitness function are tuned in an evolutionary process to obtain the classifier of the highest quality using the feature set B. The classifier quality is measured by means of the quality of new object classification. Let us finally mention recently reported results based on ensembles of classifiers constructed on the basis of different reducts (see, e.g., [57]). For more details on the application of rough sets to feature selection in a closed loop, refer to Chap. 3.

In the following sections, we point out some relationships of the rough set approach with existing methods for feature selection. The conclusion is that these methods are strongly related to extracting different kinds of reducts.

3.2 Relevance of Features

There have been both deterministic and probabilistic attempts to define *feature relevancy* [1,16,28].

Let us denote by $\mathbf{a}_i$ a vector of features (attributes),

$$(a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n),$$

obtained from the original feature vector $\mathbf{a}$ by removing a_i. By $\mathbf{v}_i$ is denoted a value of $\mathbf{a}_i$ ([16]).

A feature a_i is *relevant* if there exists some value v_i of that feature, a decision value (predictor output) v, and value $\mathbf{v}_i$ (generally a vector) for which $P(a_i = v_i) > 0$ such

that

$$P(d = v, \mathbf{a}_i = \mathbf{v}_i | a_i = v_i) \neq P(d = v, \mathbf{a}_i = \mathbf{v}_i). \tag{23}$$

In the light of this definition, a feature a_i is relevant if the probability of a **target** (given all features) can change if we remove knowledge about a value of that feature.

In [16], other definitions of *strong* and *weak relevance* were introduced.

A feature a_i is *strongly relevant* if there exists some value of that feature v_i, a value v (predictor output) of decision d and a value $\mathbf{v}_i$ of a $\mathbf{a}_i$ for which $P(a_i = v_i, \mathbf{a}_i = \mathbf{v}_i) > 0$ such that

$$P(d = v | \mathbf{a}_i = \mathbf{v}_i, a_i = v_i) \neq P(d = v | \mathbf{a}_i = \mathbf{v}_i). \tag{24}$$

Strong relevance implies that a feature is indispensable, i.e., its removal from a feature vector will change prediction accuracy.

Let us assume that $DT = (U, A, d)$ is a decision table where $V_d = \{1, \ldots, r\}$. The decision d defines the (target) decision classes $DC_s = \{x \in U | d(x) = s\}$ for $s = 1, \ldots, r$. We define a new decision table $DT_d = (U, A, d_A)$ assuming

$$d_A(x) = \left(\mu^A_{C_1}(x), \ldots, \mu^A_{C_s}(x)\right) \text{ for } x \in U. \tag{25}$$

It means that the new decision is equal to the probability distribution defined by the case (object) x in decision table DT. Now, one can show that the reducts relative to such a decision, called frequency related reducts [50], are reducts of the type discussed above.

One can also define reducts corresponding to the relevant features specified by means of the following definition of a relevant feature.

A feature a_i is *weakly relevant* if it is not strongly relevant, and there exists a subsequence $\mathbf{b}_i$ of $\mathbf{a}_i$, for which there exist some value of that feature v_i, a decision value (predictor output) v of d, and a value $\mathbf{v}_i$ of vector $\mathbf{b}_i$, for which $P(a_i = v_i, \mathbf{b}_i = \mathbf{v}_i) > 0$ such that

$$P(d = v | \mathbf{b}_i = \mathbf{v}_i, a_i = v_i) \neq P(d = v | \mathbf{b}_i = \mathbf{v}_i). \tag{26}$$

We can observe that weak relevance indicates that a feature might be dispensable (i.e., not relevant); however, sometimes (combined with some other features), it may improve prediction accuracy.

A feature is *relevant* if it is either *strongly relevant* or *weakly relevant*, otherwise, it is *irrelevant*. We can see that irrelevant features will never contribute to prediction accuracy and thus can be removed.

It has been shown in [16] that for some predictor designs, feature relevancy (even strong relevancy) does not imply that a feature must be in an optimal feature subset.

3.3 Criteria Based on Mutual Information

Entropy can be used as a *mutual information measure* of a data set for feature selection. Let us consider a decision table (data set) $DT = (U, A, d)$. Assume that $A = \{a_1, \ldots, a_n\}$. Then any *n-dimensional* pattern vector $\mathbf{a}(x) = (a_1(x), \ldots, a_n(x))$, where $x \in U$ is labeled by a decision class from $DC = (DC_1, \ldots, DC_r)$. The value of a mutual information measure for a given feature set $B \subseteq A$ can be understood as the suitability of feature subset B for classification. If initially only probabilistic knowledge about classes is given, then the uncertainty associated with the data can be measured by the entropy,

$$E(DC) = -\sum_{i=1}^{r} P(DC_i)\, log_2 P(DC_i), \tag{27}$$

where $P(DC_i)$ is the a priori probability of a class DC_i occurrence. It is known that entropy $E(DC)$ is an expected amount of information needed for class prediction.

As a measure of uncertainty, the conditional entropy $E(C|B)$ upon the subset of features B can be defined for discrete features as

$$E(DC|B) = -\sum_{all\ \mathbf{v}} P(\mathbf{v}) \left[\sum_{i=1}^{r} P(DC_i|\mathbf{v})\, log_2 P(DC_i|\mathbf{v}) \right]. \tag{28}$$

More generally, for continuous features,

$$E(DC|B) = -\int_{all\ \mathbf{v}} p(\mathbf{v}) \left[\sum_{i=1}^{r} P(DC_i|\mathbf{v})\, log_2 P(DC_i|\mathbf{v}) \right], \tag{29}$$

where $p(\mathbf{v})$ is a probability density function. The mutual information $MI(C,B)$ between the classification and feature subset B is measured by a decrease in uncertainty about the prediction of classes, given knowledge about patterns $\mathbf{v}$ formed from features B

$$J_{\text{feature}}(B) = MI(DC,B) = E(DC) - E(DC|B). \tag{30}$$

One can consider entropy related reducts [50] and Boolean reasoning to extract relevant feature sets with respect to the entropy measure. Moreover, using Boolean reasoning, one can search for frequency related reducts that preserve probability distributions to a satisfactory degree.

3.4 Criteria Based on an Inconsistency Count

An example of criteria for feature subset evaluation can be the *inconsistency measure* [24,28].

The idea of attribute reduction can be generalized by introducing a concept of *significance of attributes* that enables us to evaluate attributes not only in the two-valued scale *dispensable–relevant (indispensable)* but also in the multivalue case by assigning to an attribute a real number from the interval $[0,1]$ that expresses the importance of an attribute in the information table.

The significance of an attribute can be evaluated by measuring the effect of removing the attribute from an information table. It was shown previously that the number $\gamma(C,D)$ expresses the degree of dependency between attributes C and D or the accuracy of the approximation of U/D by C. It may now be checked how coefficient $\gamma(C,D)$ changes when attribute a is removed. In other words, what the difference is between $\gamma(C,D)$ and $\gamma(C-\{a\},D)$. The difference is normalized, and the significance of attribute a is defined by

$$\sigma_{(C,D)}(a) = \frac{\gamma(C,D) - \gamma(C-\{a\},D)}{\gamma(C,D)} = 1 - \frac{\gamma(C-\{a\},D)}{\gamma(C,D)}. \tag{31}$$

Coefficient $\sigma_{C,D}(a)$ can be understood as a classification error which occurs when attribute a is dropped. The significance coefficient can be extended to sets of attributes as follows:

$$\sigma_{(C,D)}(B) = \frac{\gamma(C,D) - \gamma(C-B,D)}{\gamma(C,D)} = 1 - \frac{\gamma(C-B,D)}{\gamma(C,D)}. \tag{32}$$

The *inconsistency rate* used in ([24]) for a reduced data set can be expressed by $J_{\text{inc}}(B) = \sigma_{(C,D)}(B)$.

Another possibility is to consider as relevant the features that come from approximate reducts of sufficiently high quality.

Any subset B of C is called an *approximate reduct* of C and the number,

$$\varepsilon_{(C,D)}(B) = \frac{\gamma(C,D) - \gamma(B,D)}{\gamma(C,D)} = 1 - \frac{\gamma(B,D)}{\gamma(C,D)}, \tag{33}$$

is called an *error of reduct approximation*. It expresses how exactly the set of attributes B approximates the set of condition attributes C with respect to determining D.

Several other methods of reduct approximation based on measures different from the positive region have been developed. All experiments confirm the hypothesis that by tuning the level of approximation the quality of the classification of new objects may be increased in most cases. It is important to note that it is once again possible to use Boolean reasoning to compute the different types of reducts and to extract relevant approximations from them.

3.5 Criteria Based on Interclass Separability

Some of the criteria for feature selection that are based on *interclass separability* are based on the idea of Fisher's linear transformation: a good feature (with high discernibility power) should cause a small within-class scatter and a large between-class scatter [9,11,13].

The rough set approach also offers methods for dealing with interclass separability. In [43], so-called *D*-reducts have been investigated. These reducts preserve not only discernibility between required pairs of cases (objects), but they also allow us to keep the distance between objects from different decision classes above a given threshold (if this is possible).

3.6 Criteria Based on a Minimum Concept Description

Open-loop type criteria of feature selection based on a minimum construction paradigm were studied [1] in machine learning and in statistics for discrete features of noise-free data sets. The straightforward technique of best feature selection could choose a minimal feature subset that fully describes all concepts (for example, classes in classification) in a given data set (see, e.g., [1,28]). Here a criterion of feature selection could be defined as Boolean function $J_{\text{feature}}(B)$ with value one if a feature subset B is satisfactory for describing all concepts in a data set; otherwise, it has a value of zero. The final selection would based on choosing a minimal subset for which a criterion gives a value of one.

The idea of feature selection, with the minimum concept description criterion, can be extended by using the concept of reduct defined in the theory of rough sets [28,44]. A reduct is a minimal set of attributes that describes all concepts. However, a data set may have many reducts. If we use the definition of the above open-loop feature selection criterion, we can see that for each reduct B, we have the maximum value of the criterion $J_{\text{feature}}(B)$. Based on a paradigm of the minimum concept description, we can select a minimum length reduct as the best feature subset. However, the minimal reduct is good for ideal situations, where a given data set fully represents a domain of interest. For real-life situations and limited-size data sets, other reducts (generally other feature subsets) might be better for generalizing prediction. A selection of a robust (generalizing) reduct, as a best open-loop feature subset, can be supported by introducing the idea of a dynamic reduct [2,4] or by an ensemble of classifiers defined by reducts [57].

3.7 Feature Selection with Individual Feature Ranking

One straightforward feature selection procedure is based on an evaluation of the predictive power of individual features, then ranking such evaluated features, and

eventually choosing the first best m features [20]. A criterion applied to an individual feature could be of either the open-loop or closed-loop type. This algorithm has limitations and assumes independence of features. It also relies on an assumption that the final selection criterion can be expressed as the sum or products of the criteria evaluated for each feature independently. It can be expected that a single feature alone may have very low predictive power, whereas this feature, when put together with others, may demonstrate significant predictive power.

One can attempt to select a minimal number $\hat{m}$ of the best ranked features that guarantees performance better or equal to a defined level according to a certain criterion $J_{\text{feature,ranked}}$. One criterion for evaluating the predictive power of a feature could be defined by the rough set *measure of significance* of the feature (attribute), discussed before.

4 Principal Component Analysis and Rough Sets for Feature Projection, Reduction, and Selection

Orthonormal projection and reduction of pattern dimensionality may improve the recognition process by considering only the most important data representation, possibly with uncorrelated elements retaining maximum information about the original data and with possible better generalization abilities.

We will discuss PCA for feature projection and reduction, followed by the joint method of feature selection using PCA and the rough set method.

4.1 Principal Component Analysis for Feature Projection and Reduction

We generally assume that our knowledge of a domain is represented as a limited-size sample of N random *n-dimensional patterns* $\mathbf{x} \in \mathbf{R}^n$ representing extracted object features. We assume that an unlabeled training data set $T = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$ can be represented as an $N \times n$ data pattern matrix $\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\right]^T$. The training data set can be characterized by the square $n \times n$ dimensional *covariance* matrix $\mathbf{R}_x$. Assume that the eigenvalues of the covariance matrix $\mathbf{R}_x$ are arranged in the decreasing order $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n \geq 0$ (with $\lambda_1 = \lambda_{max}$), with the corresponding orthonormal eigenvectors $\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^n$. Then the optimal linear transformation

$$\mathbf{y} = \hat{\mathbf{W}}\mathbf{x}, \tag{34}$$

is provided using the $m \times n$ optimal Karhunen-Loéve transformation matrix $\hat{\mathbf{W}}$ (denoted also by $\mathbf{W}_{\text{KLT}}$),

$$\hat{\mathbf{W}} = \left[\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^m\right]^T, \tag{35}$$

composed of m rows that are the first m orthonormal eigenvectors of the original data covariance matrix $\mathbf{R}_x$. The optimal matrix $\hat{\mathbf{W}}$ transforms the original *n-dimensional* patterns $\mathbf{x}$ into *m-dimensional* ($m \leq n$) feature patterns $\mathbf{y}$,

$$\mathbf{Y} = (\hat{\mathbf{W}}\mathbf{X}^T)^T = \mathbf{X}\hat{\mathbf{W}}^T, \tag{36}$$

minimizing the mean least square reconstruction error. The PCA method can be effectively used for feature extraction and dimensionality reduction by forming the *m-dimensional* ($m \leq n$) feature vector $\mathbf{y}$ containing only the first m most dominant principal components of $\mathbf{x}$. The open question remains, which principal components to select as the best for a given processing goal. One of the possible methods (criteria) for selecting a dimension of a reduced feature vector $\mathbf{y}$ is to choose a minimal number of the first m most dominant principal components $y_1, y_2, \ldots, y_m$ of $\mathbf{x}$ for which the mean square reconstruction error is less than the heuristically set error threshold ε. Another method may assume selecting the minimal number of the first m most dominant principal components for which a percentage V of a sum of unused eigenvalues of a sum of all eigenvalues,

$$V = \frac{\sum_{i=m+1}^{n} \lambda_i}{\sum_{i=1}^{n} \lambda_i} 100\%, \tag{37}$$

and is less than a defined threshold ζ.

We have applied PCA, with the resulting Karhunen–Loéve transformation (KLT) [9,11,6], for orthonormal projection (and reduction) of reduced singular value decomposition (SVD) patterns $\mathbf{x}_{\mathrm{svd},r}$ representing recognized face images.

The selection of the best principal components for classification is yet another feature selection problem. In the next section, we will discuss an application of rough sets to feature selection/reduction.

4.2 Application of Rough Set Based Reducts for Selecting of Discriminatory Features from Principal Components

The PCA, with the resulting linear Karhunen–Loéve projection, provides feature extraction and reduction optimal from the point of view of minimizing the reconstruction error. However, PCA does not guarantee that selected first principal components, as a feature vector, will be adequate for classification. Nevertheless, the projection of high-dimensional patterns into lower dimensional orthogonal principal component feature vectors might help to provide better classification for some data types.

In many applications of PCA, an arbitrary number of the first dominant principal components is selected as a feature vector. However, these methods do not cope

with the selection of the most discriminative features well suitable for classification. Even assuming that the Karhunen–Loéve projection can help in classification and can be used as a first step in the feature extraction/selection procedure, still an open question remains, "which principal components to choose for classification?"

One of the possibilities for selecting features from principal components is to apply rough set theory [28,44]. Specifically, defined in rough sets, the computation of a reduct can be used for selecting some principal components. Thus, these principal components will describe all concepts in a data set. For a suboptimal solution, one can choose the minimal length reduct or dynamic reduct as a selected set of principal components forming a selected, final feature vector. The following steps can be proposed for the PCA and rough sets based procedure for feature selection. Rough sets assume that a processed data set contains patterns labeled by associated classes with the discrete values of its elements (attributes, features). We know that PCA is predisposed to transform patterns with real-valued features (elements) optimally. Thus, after realizing the Karhunen–Loéve transformation, the resulting projected pattern features must be discretized by some adequate procedure. The resulting discrete attribute valued data set (an information system) can be processed using rough set methods.

Let us assume that we are given a limited-size data set T, containing N cases labeled by associated classes,

$$T = \{(\mathbf{x}^1, c^1_{\text{target}})(\mathbf{x}^2, c^2_{\text{target}}), \ldots, (\mathbf{x}^N, c^N_{\text{target}})\}. \tag{38}$$

Each case $(\mathbf{x}^i, c^i_{\text{target}})$ $(i = 1, 2, \ldots, N)$ is constituted of an *n-dimensional* real-valued pattern $\mathbf{x}^i \in \mathbf{R}^n$ with corresponding categorical target class c^i_{target}. We assume that a data set T contains N_i $(\sum_i^l N_i = N)$ cases from each categorical class c_i, with the total number of classes denoted by l.

Since PCA is an unsupervised method, first, from the original, class labeled data set T, a pattern part is isolated as an $N \times n$ data pattern matrix,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \ldots \\ \mathbf{x}^N \end{bmatrix}, \tag{39}$$

which each row contains one pattern. The PCA procedure is applied to the extracted pattern matrix $\mathbf{X}$, with a resulting full size an $n \times n$ optimal Karhunen–Loéve matrix $\mathbf{W}_{\text{KL}}$ (where n is the length of the original pattern $\mathbf{x}$). Now, according to the designer decision, the number $m \leq n$ of first dominant principal components has to be selected. Then, the reduced $m \times n$ Karhunen–Loéve matrix $\hat{\mathbf{W}}_{\text{KL}}$, containing only the first m rows of the full size matrix $\mathbf{W}$, is constructed. Applying the matrix $\mathbf{W}_{\text{KL}}$ the original *n-dimensional* pattern $\mathbf{x}$ can be projected, using transformation

$\mathbf{y} = \hat{\mathbf{W}}_{\mathrm{KL}}\mathbf{x}$, into the reduced *m-dimensional* pattern $\mathbf{y}$ in the principal component space. The entire projected $N \times m$ matrix $\mathbf{Y}$ of patterns can be obtained by the formula $\mathbf{Y} = \mathbf{X}\hat{\mathbf{W}}_{\mathrm{KL}}^{T}$.

At this stage, the reduced, projected data set, represented by $\mathbf{Y}$ (with real-valued attributes), has to be discretized. As a result, the discrete attribute data set represented by the $N \times m$ matrix Y_d is computed. Then, the patterns from $\mathbf{Y}_d$ are labeled by corresponding target classes from the original data set T. They form a decision table DT_m with *m-dimensional* principal component related patterns. From the decision table DT_m, one can compute the selected reduct $A_{\mathrm{feature,reduct}}$ of size l (for example, minimal length or dynamic reduct) as a final selected attribute set. Here, a reduct computation is a pure feature selection procedure.

Once the selected attribute set has been found (as a selected reduct), the final discrete attribute decision table $DT_{f,d}$ is composed. It consists of those columns from the discrete matrix $\mathbf{Y}_d$ that are included in the selected feature set $A_{\mathrm{feature,reduct}}$. Each pattern in $DT_{f,d}$ is labeled by the corresponding target class. Similarly, one can obtain a real-valued resulting reduced decision table $DT_{f,l}$ extracting (and adequately labeling by classes) those columns from the real-valued projected matrix $\mathbf{Y}$ that are included in the selected feature set $A_{\mathrm{feature,reduct}}$. Both resulting reduced decision tables can be used for classifier design.

Algorithm: Feature extraction/selection using PCA and rough sets.
Given: An *N-case* data set T containing *n-dimensional* patterns, with real-valued attributes, labeled by l associated classes $\{(\mathbf{x}^1, c^1_{target}), (\mathbf{x}^2, c^2_{target}), \ldots, (\mathbf{x}^N, c^N_{target})\}$.

1. Isolate from the original class labeled data set T a pattern part as an $N \times n$ data pattern matrix $\mathbf{X}$.
2. Compute covariance matrix $\mathbf{R}_x$ for matrix $\mathbf{X}$.
3. Compute the eigenvalues and corresponding eigenvectors for matrix $\mathbf{R}_x$, and arrange them in descending order.
4. Select the reduced dimension $m \leq n$ of a feature vector in principal component space using a defined selection method, which may be based on the judgment of the ordered values of the computed eigenvalues.
5. Compute the optimal $m \times n$ Karhunen–Loéve transform matrix $\hat{\mathbf{W}}_{\mathrm{KL}}$ based on eigenvectors of $\mathbf{R}_x$.
6. Transform original patterns from $\mathbf{X}$ into *m-dimensional* feature vectors in the principal component space by formula $\mathbf{y} = \hat{\mathbf{W}}_{\mathrm{KL}}\mathbf{x}$ for a single pattern, or formula $\mathbf{Y} = \mathbf{X}\hat{\mathbf{W}}_{\mathrm{KL}}$ for a whole set of patterns (where $\mathbf{Y}$ is an $N \times m$ matrix).
7. Discretize the patterns in $\mathbf{Y}$ with the resulting matrix $\mathbf{Y}_d$.
8. Compose the decision table DT_m constituted of the patterns from matrix $\mathbf{Y}_d$ with the corresponding classes from the original data set T.
9. Compute a selected reduct from the decision table DT_m treated as a selected set of features $A_{\mathrm{feature,reduct}}$ describing all concepts in DT_m.

10. Compose the final (reduced) discrete attribute decision table $DT_{f,d}$ containing those columns from the projected discrete matrix $\mathbf{Y}_d$ that correspond to the selected feature set $A_{\text{feature,reduct}}$. Label patterns by corresponding classes from the original data set T.
11. Compose the final (reduced) real-valued attribute decision table $DT_{f,r}$ containing those columns from the projected discrete matrix $\mathbf{Y}_d$ that correspond to the selected feature set $A_{\text{feature,reduct}}$. Label patterns by corresponding classes from the original data set T.

The results of the method of feature extraction/selection discussed depend on the data set type and three designer decisions:

1. Selection of dimension $m \leq n$ of the projected pattern in the principal component space.
2. Discretization method (and resulting quantization) of the projected data.
3. Selection of a reduct.

First, for the selected dimension m, the applied quantization method may lead to an inconsistent decision table DT_m for which no reduct exists (preserving discernibility between all pairs of objects from different decision classes). Then, a designer should return to the discretization step and select another discretization. Even if a reduct cannot be found for all possible discretization attempts, a return is realized to the stage of selecting a dimension m of the reduced feature vector $\mathbf{y}$ in the principal component space. It means that possibly the projected vector does not contain a satisfactory set of features. In this situation, a design procedure should provide the next iteration with a selected larger value of m. If a reduct cannot be found for $m = n$, a data set is not classifiable in a precise deterministic sense. Last, selection of a reduct will impact the ability of a classifier designed to generalize predictions for unseen objects.

5 Numerical Experiments — Face Recognition

As a demonstration of the role of rough set methods in feature selection/reduction, we have carried out numerical experiments of face recognition. We considered the ORL (see www.cam-orl.co.uk/facedatabase.html) face database [41] gray-scale face image data sets. We provided, separately, recognition experiments for 10 category data sets and 40 category data sets of face images. Each category was represented by 10 instances of face images. Each gray-scale face image had the dimensions of 112×92 pixels. Feature extraction from face images was provided by SVD.

Face images were classified with a single, hidden-layer error back-propagation neural network, learning vector quantization neural network (LVQ) and rule-based rough set classifier.

5.1 Singular Value Decomposition for Feature Extraction from Face Images

Singular value decomposition can be used to extract features from images [14,53]. A rectangular $n \times m$ real image represented by an $n \times m$ matrix $\mathbf{A}$, where $m \leq n$, can be transformed into a diagonal matrix by SVD. Assume that the rank of matrix $\mathbf{A}$ is $r \leq m$. The matrices $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ are nonnegative and symmetrical and have the identical eigenvalues λ_i. For $m \leq n$, there are at most $r \leq m$ nonzero eigenvalues. The SVD transform decomposes matrix $\mathbf{A}$ into the product of two orthogonal matrices, Ψ of dimension $n \times r$, and Φ of dimension $m \times r$, and a diagonal matrix $\Lambda^{1/2}$ of dimension $r \times r$. The *singular value decomposition* (SVD) of a matrix (image) $\mathbf{A}$ is given by

$$\mathbf{A} = \Psi \Lambda^{1/2} \Phi^T = \sum_{i=1}^{r} \sqrt{\lambda_i} \psi_i \varphi_i^T, \tag{40}$$

where the matrix Ψ and Φ have r orthogonal columns $\psi_i \in \mathbf{R}^n$, $\varphi_i \in \mathbf{R}^m$ $(i = 1, \ldots, r)$, respectively (representing orthogonal eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$). The square matrix $\Lambda^{1/2}$ has diagonal entries defined by

$$\Lambda^{1/2} = diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_r}), \tag{41}$$

where $\sigma_i = \sqrt{\lambda_i}$ $(i = 1, 2, \ldots, r)$ are the *singular values* of matrix $\mathbf{A}$. Each λ_i, $(i = 1, 2, \ldots, r)$ is the nonzero eigenvalue of $\mathbf{A}\mathbf{A}^T$ (as well as $\mathbf{A}^T\mathbf{A}$). Given a matrix $\mathbf{A}$ (an image) decomposed $\mathbf{A} = \Psi \Lambda^{1/2} = \Phi^T$ and since Ψ and Φ have orthogonal columns, thus the *singular value decomposition transform* (SVD transform) of the image $\mathbf{A}$ is defined as

$$\Lambda^{1/2} = \Psi^T \mathbf{A} \Phi. \tag{42}$$

If matrix $\mathbf{A}$ represents an $n \times m$ image, then r singular values $\sqrt{\lambda_i}$ $(i = 1, 2, \ldots, r)$ from the main diagonal of the matrix $\Lambda^{1/2}$ can be considered extracted features of the image. These r singular values can be arranged as an image feature vector (SVD pattern) $\mathbf{x}_{svd} = [\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_r}]^T$ of an image.

Contrary to principal component analysis, SVD is a purely matrix processing technique, not a direct statistical technique. SVD decomposition is applied to each face image separately as a face feature extraction, whereas eigenfaces [56] are obtained by projecting face vectors into principal component space derived statistically from the covariance matrix of the set of images.

Despite the expressive power of the SVD transformation [14], it is difficult to say arbitrarily how powerful the SVD features could be for classification of face images.

The *r-element* SVD patterns can be heuristically reduced by removing their r_r trailing elements whose values are below the heuristically selected threshold ε_{svd}. This can result in $n_{\text{svd},r} = r - r_r$ element reduced SVD patterns $\mathbf{x}_{\text{svd},r}$. In the next sections, we discuss techniques of finding a reduced set of face image features.

5.2 ORL Data Sets

The entire image data set was divided into training and test sets: 70% of these images were used for the training set. Given the original face image set, we applied feature extraction using SVD of matrices representing image pixels. As a result, we obtained for each image a 92-element $\mathbf{x}_{\text{svd}}$ SVD pattern where the features were the singular values of an object matrix ordered in the descending order. In the next step, we carried out several simple classification experiments using SVD patterns of different lengths to estimate the suboptimal reduction of these patterns. These patterns are obtained by cutting trailing elements from the original 92-element SVD pattern.

These experiments helped to select 60-element reduced SVD patterns $\mathbf{x}_{\text{svd},r}$. Then, according to the proposed method, we applied PCA for feature projection/reduction based on the reduced SVD patterns from the training set. Similarly to the reduction for the SVD pattern, we provided several classification experiments for different lengths of reduced PCA patterns. These patterns are obtained by considering only a selected number of the first principal components. Finally, the projected 60-element PCA patterns were in this way heuristically reduced to 20-element reduced PCA patterns $\mathbf{x}_{\text{svd},r,\text{pca},r}$. In the last preprocessing step, the rough set method was used for the final feature selection/reduction of the reduced PCA continuously valued patterns. To discretize the continuously reduced PCA features we applied the method of dividing each attribute value range into 10 evenly spaced bins. The discretized training set was used to find relevant reducts, e.g., the minimal reduct [18]. This reduct was used to form the final pattern. The training and the test sets (decision tables) with real-value pattern attributes were reduced according to the selected reduct.

In this chapter, we describe the simplest approach to relevant reduct selection. Existing rough set methods can be used to search for other forms of relevant reducts. Among them are those based on ensembles of classifiers [10]. In our approach, first a set of reducts of high quality is induced. This set is used to construct a set of predictors, and next from such predictors, the global predictor is constructed using an evolutionary approach (for details, see [57]). Predictors based on these more advanced methods make possible to achieve predictors of better quality. Certainly, the whole process of inducing such classifiers needs more time.

In all of these cases, statistical methods, e.g., cross-validation techniques, are used to estimate the robustness of the predictors constructed.

5.3 Neural Network Classifier

The error back-propagation neural network classifier designed was composed of an input layer, one hidden layer, and an output layer followed by a class-choosing module. The network learning algorithm had momentum and adaptive learning techniques built into it. First, we studied a 10-category data set with 90% of the cases

in the training set and 10% cases in the test set. We selected the five element reduct based on the reduced 20-element PCA pattern of the training set. A neural network with 50 neurons in the hidden layer was designed. The number of hidden neurons was chosen on the basis of the experiments performed. The neural network provided 99% correct classification of the test set. The rough set rule based classifier for the discretized data set restricted to the attributes from the five element reduct has exhibited 100% accuracy.

We also studied a 40-category data set with a total number of 400 cases. For this data set, we selected seven element reduct of the 320-case training set as a base for the final feature selection of reduced PCA patterns. An error back-propagation neural network with 300 neurons was designed. The number of neurons in the hidden layer was chosen experimentally. The neural network provided 96.25% correct classification of the 320-case training set and 75.5% accuracy for the 80-case test set. We applied the resilient back-propagation algorithm as a network training function that updates weight and bias values, with a performance criterion goal of 0.000299. The rough set rule based classifier for the discretized data set restricted to the attributes from the seven element reduct exhibited 94.5% accuracy for the 80-case test set.

The learning vector quantization (LVQ) neural network, trained for the training set with reduced final patterns, provided 95.8% accuracy for the test set with 28 cases. The network was trained for 200 code-book vectors and $k = 4$ neighbors.

The SVD has demonstrated a potential as a feature extraction method for face images. The processing sequence SVD, PCA with Karhunen–Loéve transformation, and the rough set approach created possibilities for a significant reduction of pattern dimensionality with an increase in classification accuracy and generalization. The classifiers considered have demonstrated the ability to recognize face images after such substantial reduction of pattern length.

6 AR Schemes and Rough Neural Networks

In the previous sections, we discussed hybrid methods for classifier construction with the application of the rough set approach, soft computing methods (e.g., neural networks), and classical statistical methods (e.g., PCA). Two basic steps can be distinguished in the methods presented: (i) reduction in preprocessing of data dimensions (number of features) and (ii) inducing classifier descriptions from reduced data. The methods are not supported by background knowledge, which could help to construct classifiers.

Now, we would like to outline an approach based on soft background knowledge represented in natural language which can be used in searching for complex classi-

fiers. We assume that background knowledge consists of some vague concept representations and relations between them, i.e., we assume that an ontology of concepts relevant to a given problem is specified. Using ontology one can derive some reasoning schemes over vague concepts. We will consider complex information granules representing approximations of such reasoning schemes. We call them AR schemes. In a distributed environment, i.e., when information about concepts is exchanged between different agents (sources of information), its is necessary to add one more component to AR schemes that is responsible for approximate translation of information granules received by agents from other agents. AR schemes extended to a distributed environment of agents are called rough neural networks.

The AR schemes are discussed in Chap. 3. We consider a special case where standards are represented by vague concepts expressed in natural language, and we outline an approach based on AR schemes. We consider applications of AR schemes for complex networks of classifiers constructed by means of experimental data and soft background knowledge.

6.1 Classifiers as Information Granules

An important class of information granules creates classifiers. One can observe that sets of decision rules generated from a given decision table $DT = (U,A,d)$ (see, e.g., [18]) can be interpreted as information granules. Classifier construction from a DT can be described as follows:

1. First, one can construct granules G_j corresponding to each particular decision $j = 1,\ldots,r$ by taking a collection $\{g_{ij} : i = 1,\ldots,k_j\}$ of left-hand sides of decision rules for a given decision.
2. Let E be a set of elementary granules (e.g., defined by conjunction of descriptors [18]) over $IS = (U,A)$. We can now consider a granule denoted by $Match(e,G_1,\ldots,G_r)$ for any $e \in E$ that is a collection of coefficients ε_{ij} where $\varepsilon_{ij} = 1$ if the set of objects defined by e in IS is included in the meaning of g_{ij} in IS, i.e., $Sem_{IS}(e) \subseteq Sem_{IS}(g_{ij})$; and zero, otherwise. Hence, the coefficient ε_{ij} is equal to one if and only if granule e matches granule g_{ij} in IS.
3. Let us now denote by $Conflict\_res$ an operation (resolving conflict between decision rules recognizing elementary granules) defined on granules of the form $Match(e,\ G_1,\ \ldots,\ G_r)$ with values in the set of possible decisions $1,\ldots,r$. Hence,

$$Conflict\_res\,[Match(e,G_1,\ldots,G_r)]\,,$$

is equal to the decision predicted by the classifier,

$$Conflict\_res\,[Match(\bullet,G_1,\ldots,G_r)]\,,$$

on the input granule e.

Hence, classifiers are special cases of information granules. Parameters to be tuned are voting strategies, matching strategies of objects against rules as well as other parameters like closeness of granules in the target granule.

Classifier construction is illustrated in Fig. 1, where three sets of decision rules are presented for the decision values 1,2, and 3, respectively. Hence, $r = 3$. In the figure, to omit too many indexes, we write α_i instead of g_{i1}, β_i instead of g_{i2}, and γ_i instead of g_{i3}, respectively. Moreover, $\varepsilon_1, \varepsilon_2, \varepsilon_3$, denote $\varepsilon_{1,1}, \varepsilon_{2,1}, \varepsilon_{3,1}$; $\varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7$ denote $\varepsilon_{1,2}, \varepsilon_{2,2}, \varepsilon_{3,2}, \varepsilon_{4,2}$; and $\varepsilon_8, \varepsilon_9$ denote $\varepsilon_{1,3}, \varepsilon_{2,3}$, respectively. The reader can

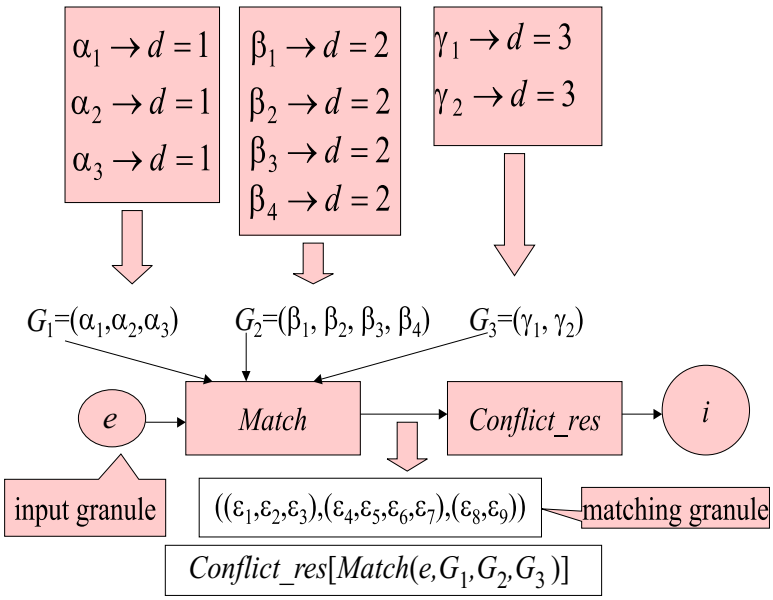

Fig. 1. Classifiers as information granules

now easily describe more complex classifiers by means of information granules. For example, one can consider soft instead of crisp inclusion between elementary information granules representing classified objects and the left-hand sides of decision rules or soft matching between recognized objects and the left-hand sides of decision rules.

6.2 Soft Background Knowledge

We are assuming a knowledge base formulated by means of soft concepts, and relations between them are given. Such background knowledge is called soft ontology and is represented in natural language. One can construct derivations over such ontologies. They represent reasoning schemes in natural language. We are interested in

derivations with conclusions representing decision classes. Such derivations can be treated as soft descriptions of cases. A set of such derivations is called a knowledge base. It consists of soft information about conclusions related to complex concepts, assuming that some simpler or elementary ones are satisfied. We are going to show how such knowledge bases can be used in searching for complex classifiers. Moreover, we present an outline for carrying out reasoning from measurements to conclusions about complex soft concepts using AR schemes and rough neural networks. One can treat our approach as a search method for relevant features supported by background knowledge represented in natural language. Our methodology can be treated as one for building interfaces between experimental knowledge and expert knowledge represented in natural language. The aim is to use the background knowledge to derive conclusions from experimental data.

6.3 Construction of Complex Classifiers from Simpler Ones Using Soft Background Knowledge

In this section, we discuss the possibility of using a soft knowledge base as a guide in searching for relevant features for constructing more complex classifiers from simpler ones. Any soft rule from a soft knowledge base with a left-hand side consisting of a conjunction of soft conditions (representing soft concepts) and the right-hand side consisting of target condition (representing the target soft concept) can be used for construction of a classifier for a target concept from classifiers for conditions. We assume that classifiers for conditions are induced. Hence, relevant features for approximating these concepts are encoded in these classifiers. However, for the target concept, we know only the sample of objects with corresponding decisions without relevant features for classification or recognition. Our assumption about the rules in a knowledge base is that the relevant features for a target concept can be discovered in feature spaces *that are not far* from feature spaces of condition classifiers. Using Boolean reasoning one can measure the distance between feature spaces by means of the complexity of the construction that it is necessary to perform to reach one such space from another. We would like to illustrate this intuition by presenting several examples of relevant feature spaces for the target concept's approximate description. Such feature spaces can include features described by

1. conjunctions of descriptors;
2. disjunctions of conjunctions of descriptors;
3. disjunctions of conjunctions of descriptor disjunctions; and
4. characteristic functions of clusters.

In all cases, the descriptors are selected from the feature spaces of input classifiers from which a new target classifier is constructed. One can observe that features described by descriptor disjunctions correspond to a symbolic value grouping of nominal features or a discretization of real value features [18]. Disjunctions of conjunctions of such features describe higher level patterns generalized next to clusters.

The clusters are constructed by means of such patterns and by an appropriately chosen similarity measure between patterns. The similarity measure should make it possible to generalize the previously defined patterns to clusters. The clusters are next used for defining features relevant to the new classifier construction. A mechanism for measuring the degree of closeness of input patterns to such clusters should be developed for computing of degrees to which analyzed objects are included in the cluster. Such degrees are treated as values of features defined by clusters.

One can interpret the process of searching for relevant features as a searching process for descriptors corresponding to such clusters. Such descriptors should satisfy the following constraint: sets of objects defined by descriptor conjunctions should be included to a satisfactory degree in a given concept (e.g., decision class) and should be supported by sufficiently many objects. In this way, such descriptors are making it possible to obtain short descriptions of concept approximations.

Certainly, one can use some more sophisticated operations transforming the feature spaces of condition classfiers into feature spaces of target classifiers. Evolutionary computing [19] can search for relevant features in such feature spaces.

The basic assumption is that using the soft knowledge base can help us to discover relevant features for more complex classifiers.

The approach discussed does not yet guarantee the robustness of classifiers, i.e., preserving the high quality of new object classification (or recognition) under acceptable deviations of information about objects. We propose an approach making it possible to eliminate this drawback. The approach is based on methods for constructing reasoning clusters constructed *along* derivations in natural language. These reasoning clusters link pattern clusters consisting of patterns sufficiently included in so-called standards (prototypes) or close to each other. The inclusion (closeness) degree of patterns in clusters is controlled to guarantee that under deviations of input patterns, the deviation of output patterns still returns acceptable solutions. This idea is formalized by using AR schemes and rough neural networks. In the following section, we outline a solution based on AR schemes and rough neural networks, and we emphasize their possible applications in pattern recognition.

6.4 AR Schemes and Rough-Neural Networks

In this section, we briefly recall an approach for approximate reasoning based on AR schemes (see Chap. 3). We use terminology from the multiagent area [15].

We assume each agent $ag \in Ag$ is equipped with a system of information granules $S(ag)$. Using such a system, the agent ag creates a representation for all of its components.

Agents are able to extract local approximate reasoning schemes, called productions, from such representations. Algorithmic methods for extracting such productions from data are discussed, e.g., in [30,45,47]. They are based on decomposition strategies.

The right-hand side of any *production* for decomposition of condition α at *ag* is of the form

$$\alpha, \varepsilon^{(i)}, \tag{43}$$

and the left-hand side is of the form

$$\alpha_1, \varepsilon_1^{(i)}; \ldots; \alpha_n, \varepsilon_n^{(i)}, \tag{44}$$

where $i = 1, \ldots, k$ for some k.

Such a production represents information about an operation o that can be performed by an agent *ag*. In the production, n denotes the arity of operation. The operation o represented by the production transforms standard (prototype) input information granules represented by $\alpha_1, \ldots, \alpha_n$ into standard (prototype) information granule α. Moreover, if input information granules $g_1, \ldots, g_n$ are included (close) to $\alpha_1, \ldots, \alpha_n$ to degrees $\varepsilon_1^{(i)}, \ldots, \varepsilon_n^{(i)}$, then the result of operation o on information granules $g_1, \ldots, g_n$ is included (close) in the standard α to a degree at least $\varepsilon^{(i)}$, where $1 \leq i \leq k$. Standard (prototype) granules can be interpreted in different ways. In particular, in the applications discussed for pattern recognition, they describe the centers of discovered clusters. In more general cases, standards correspond to concept names expressed in natural language.

Sample productions are basic components of a reasoning system related to the agent set *Ag*. An important property of such productions is that they are expected to be discovered from available experimental data and background knowledge. Let us observe also that the degree structure is not necessarily restricted to positive reals from the interval $[0,1]$. The inclusion degrees can be complex information granules used to represent the degree of inclusion.

It is worthwhile mentioning that productions can also be interpreted as constructive descriptions of some operations on fuzzy sets (see Chap. 3). The methods for such constructive descriptions are based on rough sets and Boolean reasoning (see, e.g., [18,28]).

Reasoning in multiagent system can be represented as a process of constructing information granules. This process is not restricted to internal operations performed by agents. The agents can communicate. In this process, they exchange some information granules. It is important to note that any agent possesses her/his own information granule system. Hence, a granule received by one agent from another

agent cannot be, in general, understood precisely by the receiving agent. We assume that associated with the jth argument of any operation o performed by an agent ag, there is an approximation space $AS(ag)^j$ (see, e.g., [38,47]) making it possible to construct relevant approximations of the received information granules used next as operation arguments. The result of approximation is an information granule in the information granule system of agent ag. In some cases, approximation can be induced using rough set methods (see, e.g., [47]). In general, constructing information granule approximations is a complex process because, for instance, a high-quality approximation of concepts often can be obtained only through dialog (including negotiations, conflict resolution, and cooperation) among agents. In this process, the approximation can be constructed gradually when dialog is progressing.

Approximation spaces are usually parameterized. This means that it is necessary to tune their parameters to find suboptimal approximations of information granules. This observation was the starting point for the rough-neurocomputing paradigm (see [26,38,45,47] and Chap. 3).

In general, the inputs of rough neurons are derived from information granules instead of real numbers, and parameterized approximation spaces correspond to real weights in the classical neuron. The result of operation o depends on the parameters chosen for approximation spaces. The process of tuning the parameters of such approximation spaces corresponds to the process of weight tuning in classical neurons (see Fig. 1 in Chap. 2).

Now, we are able to discuss one of the main concepts of our approach, approximate reasoning schemes (AR schemes). They can be treated as derivations obtained by using the productions of different agents. Assume, for simplicity of consideration, that agents are working using the same system of information granules, i.e., they do not use approximation spaces to approximate granules received from other agents. The approach can be extended to the more general case. The relevant derivations defining AR schemes satisfy a so-called robustness (or stability) condition, that is, at any node of a derivation, the inclusion (or closeness) degree of a constructed granule (to a given standard) is higher than required by the production to which the result should be sent. This makes it possible to obtain a sufficient robustness condition for the whole derivation. For details refer to [31,34–37] and to chapters in this book discussing the foundations of rough-neurocomputing approach. In the general case, i.e., when it is necessary to use approximation spaces, the AR schemes can be interpreted as rough neural networks. When standards are interpreted as concept names in natural language and there is given a reasoning scheme in natural language over such standards, the corresponding rough neural network represents a cluster of reasoning constructions approximately following (in other information granule systems) the reasoning given in natural language.

Let us observe that AR schemes are not classical proofs defined by means of de-

ductive systems. They are approximate reasoning schemes discovered from data and background knowledge. The notion of classical proof is substituted by derivations defining AR schemes, i.e., derivations satisfying some constraints. Deductive systems are substituted by productions systems of agents linked by approximation spaces, communication strategies, and mechanisms deriving AR schemes. This revision of classical logical notions seems to be important for solving complex pattern recognition problems.

6.5 Illustrative Example

Let us consider a very simple illustrative example of the face-recognition problem. Assume that among the concepts of a given knowledge base are the following concepts:

1. $exactly\_one\_ear\_visible = yes$.
2. $nose\_shape\_visible = yes$.
3. $nose\_shape\_in\_front\_view = sharp$.
4. $nose\_shape = sharp$.
5. $face\_in\_side\_view = yes$.

and rules

1. **If** $exactly\_one\_ear\_visible = yes$ **and** $nose\_shape\_visible = yes$,
   then $face\_in\_side\_view = yes$.
2. **If** $face\_in\_side\_view = yes$
   and $nose\_shape = sharp$,
   then $nose\_shape\_in\_front\_view = sharp$.

First, a classifier for the concept

$$face\_in\_side\_view = yes,$$

is constructed in the context of classifiers for

$$exactly\_one\_ear\_visible = yes, \quad nose\_shape\_visible = yes,$$

and next a classifier for $nose\_shape\_in\_front\_view = sharp$ is constructed in the context of its sensory classifiers,

$$face\_in\_side\_view = yes, \quad nose\_shape = sharp.$$

Then, productions for such rules are induced. Finally, they are used to derive robust AR schemes. From such schemes, one can predict on the basis of estimates from sensory classifiers that the *nose_shape_in_front_view* should be to a high degree *sharp* for a given object *x* if sensory properties for this object *x* are satisfied to

sufficient degrees:

exactly_one_ear_visible = yes, *nose_shape_visible = yes*, and *nose_shape = sharp*.

This can be confronted with another conclusion of approximate reasoning on objects from the database, e.g., the face is to a sufficient degree *in_the_front_view* and the *nose_shape* is to a sufficient degree *non_sharp*. Such objects can be eliminated from candidates identifying x in the database.

7 Conclusions

We have presented a rough set method and its role in feature selection for pattern recognition.

In the first part, we proposed a sequence of data mining steps, including application of SVD, PCA, and rough sets, for feature selection. This processing sequence has shown a potential for feasible feature extraction and feature selection in designing neural network classifiers for face images. The method discussed provides a substantial reduction of pattern dimensionality. Rough set methods have shown the ability to reduce significantly pattern dimensionality and have proven to be viable data mining techniques as the front end of neural network classifiers.

In the second part, we discussed an approach to pattern recognition based on rough-neurocomputing with the application of soft knowledge bases. This research direction seems to be promising for complex pattern recognition problems, such as identification of objects and path planning by autonomous systems.

Acknowledgments
The research has been partially supported by the COBASE project from NSF National Research Council, USA, National Academy of Sciences, USA and Poland 2000–2001. Moreover, the research of Andrzej Skowron has been partially supported by the State Committee for Scientific Research of the Republic of Poland (KBN), research grant 8 T11C 025 19, and by a Wallenberg Foundation grant.

References

1. H. Almuallim, T.G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 574–552, AAAI Press, Menlo Park, CA, 1991.
2. J. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision system. In *[32]*, 321–365, 1998.
3. J. Bazan, S.H. Nguyen, H.S. Nguyen, P. Synak, J. Wróblewski. Rough set algorithms in classification problems. In *[29]*, 49–88, 2000.

4. J. Bazan, A. Skowron, P. Synak. Dynamic reducts as a tool for extracting laws from decision tables. In *Proceedings of the Symposium on Methodologies for Intelligent Systems (ISMIS'94)*, LNAI 869, 346–355, Springer, Berlin, 1994.
5. J. Bazan, A. Skowron, P. Synak. *Market data analysis: A rough set approach*. Report number 6 of the Institute of Computer Science, Warsaw University of Technology, 1994.
6. C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.
7. G. Boole. *An Investigation of the Laws of Thought on which are Founded the Mathematical Theories of Logic and Probabilities*. Walton and Maberley, London, 1854.
8. F.M. Brown. *Boolean Reasoning*. Kluwer, Dordrecht, 1990.
9. K. Cios, W. Pedrycz, R. Swiniarski. *Data Mining Methods for Knowledge Discovery*. Kluwer, Boston, 1998.
10. T.G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18(4): 97–136, 1997.
11. R.O. Duda, P.E. Hart. *Pattern Recognition and Scene Analysis*. Wiley, New York, 1973.
12. I. Duentsch, G. Gediga. Statistical evaluation of rough set dependency analysis. *International Journal of Human–Computer Studies*, 46: 589–604, 1997.
13. K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.
14. Z.Q. Hong. Algebraic feature extraction of image for recognition. *Pattern Recognition*, 24(3): 211–219, 1991.
15. M.N. Huhns, M.P. Singh, editors. *Readings in Agents*. Morgan Kaufmann, San Mateo, CA, 1998.
16. G. John, R. Kohavi, K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the 11th International Conference (ICML'94)*, 121–129, Morgan Kaufmann, San Mateo, CA, 1994.
17. J. Kittler. Feature selection and extraction. In T.Y. Young, K.S. Fu, editors, *Handbook of Pattern Recognition and Image Processing*, 59–83, Academic Press, New York, 1986.
18. J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron. Rough sets: A tutorial. In *[27]*, 3–98, 1999.
19. J. Koza, editor. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
20. M. Kudo, J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33: 25–41, 2000.
21. P. Langley, S. Sage. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, 140–144, AAAI Press, Menlo Park, CA, 1994.
22. H. Liu, H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Approach*. Kluwer, Boston, 1998.
23. H. Liu, H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer, Boston, 1998.
24. H. Liu, R. Setiono. A probabilistic approach to feature selection - a filter solution. In *Proceedings of the 13th International Conference on Machine Learning (ICML'96)*, 319–327, Springer, Heidelberg, 1996.
25. V. Lobo, F. Moura-Pires, R. Swiniarski. *Minimizing the number of neurons for a SOM-based classification, using Boolean function formalization*. Report number 08/4/97 of Department of Mathematical and Computer Sciences, San Diego State University, San Diego, CA, 1997.
26. S.K. Pal, W. Pedrycz, A. Skowron, R. Swiniarski, editors. Rough-neuro computing (special issue). Vol. 36 of *Neurocomputing: An International Journal*, 2001.

27. S.K. Pal, A. Skowron, editors. *Rough Fuzzy Hybridization: A New Trend in Decision–Making*. Springer, Singapore, 1999.
28. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.
29. L. Polkowski, Y.Y. Lin, S. Tsumoto, editors. *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*. Physica, Heidelberg, 2000.
30. L. Polkowski, A. Skowron. Rough mereological approach to knowledge–based distributed AI. In J.K. Lee, J. Liebowitz, J.M. Chae, editors, *Proceedings of the 3rd World Congress on Expert Systems*, 774–781, Cognizant Communication Corporation, New York, 1996.
31. L. Polkowski, A. Skowron. Rough mereological foundations for design, analysis, synthesis, and control in distributed systems. *Information Sciences An International Journal*, 104(1-2): 129–156, 1998.
32. L. Polkowski, A. Skowron, editors. *Rough Sets in Knowledge Discovery 1: Methodology and Applications*. Physica, Heidelberg, 1998.
33. L. Polkowski, A. Skowron, editors. *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*. Physica, Heidelberg, 1998.
34. L. Polkowski, A. Skowron. Grammar systems for distributed synthesis of approximate solutions extracted from experience. In G. Paun, A. Salomaa, editors, *Grammar Models for Multiagent Systems*, 316–333, Gordon and Breach, Amsterdam, 1999.
35. L. Polkowski, A. Skowron. Towards adaptive calculus of granules. In L.A. Zadeh, J. Kacprzyk, editors, *Computing with Words in Information/Intelligent Systems 1*, 201–227, Physica, Heidelberg, 1999.
36. L. Polkowski, A. Skowron. Rough mereology in information systems. A case study: Qualitative spatial reasoning. In *[29]*, 89–135, 2000.
37. L. Polkowski, A. Skowron. Rough mereological calculi of granules: A rough set approach to computation. *Computational Intelligence*, 17(3): 472–492, 2001.
38. L. Polkowski, A. Skowron. Rough-neuro computing. In W. Ziarko, Y.Y. Yao, editors, *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC 2000)*, LNAI 2005, 57–64, Springer, Berlin, 2001.
39. J.R. Quinlan, editor. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
40. J. Rissanen. Modeling by shortest data description. *Automatica*, 14: 465–471, 1978.
41. F. Samaria, A. Harter. Parameterization of stochastic model for human face identification. In *Proceedings of IEEE Workshop on Application of Computer Vision*, 1994. Available at www.cam-orl.co.uk/facedatabase.html.
42. B. Selman, H. Kautz, A. McAllester. Ten challenges in propositional reasoning and search. In *Proceedings of IJCAI'97*, 50–54, Morgan Kaufmann, San Francisco, 1997.
43. A. Skowron. Extracting laws from decision tables. *Computational Intelligence*, 11(2): 371–388, 1995.
44. A. Skowron. Rough sets in KDD. In Z. Shi, B. Faltings, M. Musem, editors, *16th World Computer Congress (IFIP 2000):Proceedings of Conference on Intelligent Information Processing (IIP 2000)*, 1–17, Publishing House of Electronic Industry, Beijing, 2000 (plenary talk).
45. A. Skowron. Toward intelligent systems: Calculi of information granules. In S. Hirano, M. Inuiguchi, S. Tsumoto, editors, *Proceedings of International Workshop on Rough Set Theory and Granular Computing (RSTGC-2001)*, Vol. 5(1/2) of *Bulletin of International Rough Set Society*, 9–30, 2001 (keynote speech).

46. A. Skowron, C. Rauszer. The discernibility matrices and functions in information systems. In *[51]*, 331–362, 1992.
47. A. Skowron, J. Stepaniuk. Information granules: Towards foundations of granular computing. *International Journal of Intelligent Systems*, 16(1): 57–86, 2001.
48. D. Ślęzak. Approximate reducts in decision tables. In *Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, Vol. 3, 1159–1164, Universidad da Granada, Granada, 1996.
49. D Ślęzak. Various approaches to reasoning with frequency based decision reducts: A survey. In *[29]*, 235–285, 2000.
50. D. Ślęzak. *Approximate Decision Reducts*. Ph.D. Dissertation, Faculty of Mathematics, Informatics and Mechanics, Warsaw University, 2002 (in Polish).
51. R. Słowiński, editor. *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer, Dordrecht, 1992.
52. R. Swiniarski. Introduction to rough sets. In *Materials of The International Short Course on Neural Networks, Fuzzy and Rough Systems. Theory and Applications*, 1–24, San Diego State University Press, San Diego, CA, 1993.
53. R. Swiniarski, J. Nguyen. Rough sets expert system for texture classification based on 2D spectral features. In *Proceedings of the 3rd Biennial European Joint Conference on Engineering Systems Design and Analysis (ESDA'96)*, 3–8, Montpellier, France, 1996.
54. R. Swiniarski, F. Hunt, D. Chalvet, D. Pearson. Feature selection using rough sets and hidden layer expansion for rupture prediction in a highly automated production system. In *Proceedings of the 12th International Conference on Systems Science*, 12–15, Wrocław, Poland, 1995.
55. M. Szczuka. *Neural Networks and Symbolic Methods for Classifier Construction*. Ph.D. Dissertation, Faculty of Mathematics, Informatics and Mechanics, Warsaw University, 2000 (in Polish).
56. M.A. Turk, A.P. Pentland. Face recognition using eigenspaces. In *Proceedings of the 1991 IEEE Conference on Vision and Pattern Recognition (CVPR'91)*, 586–591, Maui, Hawaii,1991.
57. J. Wróblewski. *Adaptive Methods for Object Classification*. Ph.D. Dissertation, Faculty of Mathematics, Informatics and Mechanics, Warsaw University, 2002 (in Polish).
58. L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4: 103–111, 1996.