Int. J. Appl. Math. Comput. Sci., 2001, Vol.11, No.3, 565–582

ROUGH SETS METHODS IN FEATURE REDUCTION AND CLASSIFICATION

ROMAN W. ŚWINIARSKI*

The paper presents an application of rough sets and statistical methods to feature reduction and pattern recognition. The presented description of rough sets theory emphasizes the role of rough sets reducts in feature selection and data reduction in pattern recognition. The overview of methods of feature selection emphasizes feature selection criteria, including rough set-based methods. The paper also contains a description of the algorithm for feature selection and reduction based on the rough sets method proposed jointly with Principal Component Analysis. Finally, the paper presents numerical results of face recognition experiments using the learning vector quantization neural network, with feature selection based on the proposed principal components analysis and rough sets methods.

Keywords: rough sets, feature selection, classification

1. Introduction

One of the fundamental steps in classifier design is reduction of pattern dimensionality through feature extraction and feature selection (Cios *et al.*, 1998; Kittler, 1986; Langley and Sage, 1994; Liu and Motoda, 1999). Feature selection is often isolated as a separate step in the processing of pattern sets.

Features may be irrelevant (having no effect on the processing performance) or relevant (having an impact on the processing performance). A feature may have a different discriminatory or predictive power. We present rough sets methods and Principal Components Analysis in the context of feature selection in pattern classification. At the beginning, the paper presents an introduction to rough sets theory (Pawlak, 1991; Skowron, 1990) and its role in feature selection. Then, we present a short overview of the feature selection problem, including the open-loop and the closed-loop feature selection methods (Cios *et al.*, 1998). This section focuses the discussion on feature selection criteria including rough set-based methods. The next section presents a short description of the Principal Component Analysis (PCA) (Cios *et al.*, 1998) as a method of feature projection and reduction. It also contains a description of rough set-based methods, proposed jointly with PCA, for feature projection and reduction. The following section describes results of numerical experiments of

* San Diego State University, Department of Mathematical and Computer Sciences, 5500 Campanile Drive, San Diego, CA 92182, U.S.A., e-mail: rswiniar@sciences.sdsu.edu

face recognition using the presented rough set-based method of feature selection and Learning Vector Quantization neural networks. This section also contains a short description of feature extraction from facial images using Singular Value Decomposition (SVD).

2. Rough Sets and Feature Selection

Rough sets theory has been proposed by Professor Pawlak for knowledge discovery in databases and experimental data sets (Pawlak, 1982; 1991; Skowron, 1990). It is based on the concept of an *upper* and a *lower approximation* of a set, the *approximation space* and models of sets.

An *information system* can be represented as

$$S = \langle U,\ Q,\ V,\ f \rangle, \tag{1}$$

where U is the *universe*, a finite set of N objects $\{x_1, x_2, \ldots, x_N\}$ (a nonempty set), Q is a finite set of *attributes*, $V = \cup_{q \in Q} V_q$ (where V_q is a *domain* of the attribute q), $f : U \times Q \rightarrow V$ is the total *decision function* (called the information function) such that $f(x, q) \in V_q$ for every $q \in Q,\ x \in U$. A subset of attributes $A \subseteq Q$ defines an *equivalence relation* (called an *indiscernibility relation*) on U

$$IND(A) = \big\{(x,\ y) \in U :\ \text{for all}\ \ a \in A,\ f(x,\ a) = f(y,\ a)\big\}, \tag{2}$$

denoted also by $\tilde{A}$. The information system can also be defined as a *decision table*

$$DT = \langle U,\ C \cup D,\ V,\ f \rangle, \tag{3}$$

where C is a set of *condition* attributes, D is a set of *decision* attributes, $V = \bigcup_{q \in C \cup D} V_q$, where V_q is the set of the *domain* of an attribute $q \in Q$, $f : U \times (C \cup D) \rightarrow V$ is a total *decision function* (information function, decision rule in DT) such that $f(x,\ q) \in V_q$ for every $q \in Q$ and $x \in V$.

For a given S a subset of attributes $A \subseteq Q$ determines the approximation space $AS = (U, IND(A))$ in S. For given $A \subseteq Q$ and $X \subseteq U$ (a concept X), the *A-lower approximation* $\underline{A}X$ of the set X in AS and the *A-upper approximation* $\bar{A}X$ of the set X in AS are defined as follows:

$$\underline{A}X = \{x \in U : [x]_A \subseteq X\} = \bigcup \{Y \in A^* : Y \subseteq X\}, \tag{4}$$

$$\bar{A}X = \{x \in U : [x]_A \cap X \neq \emptyset\} = \bigcup \{Y \in A^* : Y \cap X \neq \emptyset\}. \tag{5}$$

Certain attributes in an information system may be redundant and can be eliminated without losing essential classificatory information. One can consider feature (attribute) reduction as the process of finding a smaller (than the original one) set of attributes with the same or close classificatory power as the original set. Rough sets provide a method to determine for a given information system the most important attributes from a classificatory power point of view. The concept of the *reduct* is fundamental for rough sets theory. A reduct is the essential part of an information

system (related to a subset of attributes) which can discern all objects discernible by the original set of attributes of an information system. Another important notion relates to a *core* as a common part of all reducts. The core and reduct are important concepts of rough sets theory that can be used for feature selection and data reduction.

Rough sets theory determines a degree of attributes' dependency and their significance. For the information system $S = \langle U,\ Q,\ V,\ f\rangle$, with condition and decision attributes $Q = C \cup D$, for a given set of condition attributes $A \subset C$, we can define the *A-positive region* $POS_A(D)$ in the relation $IND(D)$ as

$$POS_A(D) = \bigcup \{\underline{A}X : X \in IND(D)\}. \tag{6}$$

The positive region $POS_A(D)$ contains all the objects in U that can be classified without an error into distinct classes defined by $IND(D)$, based only on information in the relation $IND(A)$. One can form a positive region for any two subsets of attributes $A, B \in Q$ in the information system S. Since the subset of attributes $B \in Q$ defines the indiscernibility relation $IND(B)$, it consequently defines the classification B^* $(U/IND(B))$ with respect to the subset A. The *A-positive region* of B is defined as

$$POS_A(B) = \bigcup_{X \in B^*} \underline{A}X. \tag{7}$$

The A-positive region of B contains all the objects that, by using attributes A, can be certainly classified to one of distinct classes of the classification B^*.

The cardinality of the A-positive region of B can be used to define a measure (a degree) $\gamma_A(B)$ of dependency of the set of attributes B on A:

$$\gamma_A(B) = \frac{\mathrm{card}(POS_A(B))}{\mathrm{card}(U)}. \tag{8}$$

In the information system S a set of attributes B *depends* (is *dependent*) on a set A in S, which is denoted by $A \to B$, iff an equivalence relation satisfies $IND(A) \subseteq IND(B)$. Two sets A and B are *independent* in S iff neither $A \to B$ nor $B \to A$ holds. The dependency of set B to *degree* k to the set A in S is denoted as follows:

$$A \xrightarrow{k} B, \quad 0 \le k \le 1, \text{ if } k = \gamma_A(B), \tag{9}$$

where $\gamma_A(B)$ was described above.

Rough sets define a *measure of significance* (*coefficient of significance*) of the attribute $a \in A$ from the set A with respect to the classification B^* $(U/IND(B))$ generated by a set B:

$$\mu_{A,B}(a) = \frac{\mathrm{card}\,(POS_A(B)) - \mathrm{card}\,(POS_{A-\{a\}}(B))}{\mathrm{card}\,U}. \tag{10}$$

A significance of the attribute a in the set $A \subseteq Q$, computed with respect to the original classification Q^* (generated by the entire set of attributes Q from the information system S), can be denoted by

$$\mu_A(a) = \mu_{A,Q}(a). \tag{11}$$

Reduct and Core

Some attributes of an information system may be redundant (superfluous) with respect to a specific classification A^* generated by attributes $A \subseteq Q$. By virtue of the dependency properties of attributes, one can find a *reduced* set of the attributes by removing *superfluous* attributes, without loss of the classification power of the reduced information system. For a given information system S and a subset of attributes $A \subseteq Q$, an attribute $a \in A$ is called *dispensable* in the set A if $IND(A) = IND(A - \{a\})$ (it means that indiscernibility relations generated by sets A and $A - \{a\}$ are identical). Otherwise the attribute a is *indispensable* in A. It can be found that the dispensable attribute does not improve the classification of the original information system S (the attribute is irrelevant). The set of all indispensable attributes in the set $A \subseteq Q$ is called the *core* of A in S, and it is denoted by $CORE(A)$. The core contains all the attributes that cannot be removed from the set A without changing the original classification A^*.

Let us consider two subsets of attributes $A, B \subseteq Q$ in S. An attribute a is called B-*dispensable* (indispensable with respect to B) in the set A if $POS_A(B) = POS_{A-\{a\}}(B)$. Otherwise the attribute a is B-*indispensable*. If every attribute of A is B-indispensable, then A is indispensable with respect to B. In rough sets theory the set of all B-indispensable attributes from the set A is called a B-*relative core* (or B-*core*) of A, and it is denoted by $CORE_B(A)$,

$$CORE_B(A) = \big\{a \in A \,:\, POS_A(B) \neq POS_{A-\{a\}}(B)\big\}. \tag{12}$$

The set $A \subseteq Q$ is called *orthogonal* if all its attributes are indispensable. A proper subset $E \subset A$ is defined as a *reduct* set of A in S if E is orthogonal and preserves the classification generated by A. Hence a reduct set of A, denoted by $RED(A)$, is defined as

$$E = RED(A) \Longleftrightarrow \big(E \subset A,\ IND(E) = IND(A),\ E \text{ is orthogonal}\big), \tag{13}$$

where E is a reduct of A (i.e. $E = RED(A)$ if E is a minimal set of attributes which discerns all the objects in S discernible by the whole set A, and which cannot be further reduced. All the reducts (family of reducts) of A are denoted by $RED^F(A)$. We see that the intersection of all the reducts of A is a core of A:

$$CORE(A) = \bigcap RED(A). \tag{14}$$

Relative Reduct

Rough sets define also a *relative reduct* related to two sets of attributes $A, B \subseteq Q$ in S. The set A is called B-*orthogonal* if all the attributes of A are B-indispensable. A B-orthogonal proper subset of A is called a B-*reduct* of A and it is denoted by $RED_B(A)$:

$$E = RED_B(A) \Longleftrightarrow (E \subset A,\ POS_E(B) = POS_A(B),\ E \text{ is } B\text{-orthogonal}) \tag{15}$$

The subset $E \subset A$ is called a B-*reduct* of A in S if E is independent of B and $POS_E(B) = POS_A(B)$. A B-reduct $RED_B(A)$ of A is a minimal set of attributes in A which discern all the objects in S discernible by the entire set A, and which

cannot be further reduced. All B-reducts (family) are denoted by $RED_B^F(A)$. The intersection of all B-reducts of A is the *relative B-core* of A,

$$CORE_B(A) = \bigcap RED_B(A). \tag{16}$$

3. Feature Selection

One can consider *feature selection* as the process of finding a subset of features, from the original set of pattern features, optimally according to the defined criterion. Consider a data set T_{all} (containing N_{all} cases), constituted with n-feature patterns $\mathbf{x}$ (labeled or unlabeled by target values). Let all n features of a pattern form a whole original feature set $X_{\text{all}} = \{x_1, x_2, \ldots, x_n\}$. An optimal feature selection is the process of finding a subset $X_{\text{opt}} = \{x_{1,\text{opt}}, x_{2,\text{opt}}, \ldots, x_{m,\text{opt}}\}$ containing $m \leq n$ features from the set of all original features $X_{\text{opt}} \subseteq X_{\text{all}}$, which guarantees the accomplishment of a processing goal while minimizing a defined feature selection criterion $J_{\text{feature}}(X_{\text{feature\_subset}})$.

3.1. Rough Sets and Relevance of Features

The feature relevance can be interpreted using rough sets theory (Pawlak, 1991; Pal and Skowron, 1999, Cios *et al.*, 1998). The probabilistic and deterministic definitions of feature relevance were presented in (Almuallim and Dietterich, 1991; John *et al.*, 1994; Pawlak, 1991). Let us assume that we are given a class-labeled data set T with N cases $(\mathbf{x}, \mathbf{target})$, containing n-feature patterns $\mathbf{x}$ and associated **targets**. Let us introduce a vector of features $\mathbf{v}_i = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)^T$ (with its values denoted by $\mathbf{a}_{\mathbf{v}_i}$) obtained from an original feature vector $\mathbf{x}$ by removing the x_i feature (John *et al.*, 1994).

A feature x_i is *relevant* if there exists some value of that feature a_{x_i} and a predictor output value $\mathbf{a}_\mathbf{y}$ (generally a vector) for which $P(x_i = a_{x_i}) > 0$ such that

$$P\big(\mathbf{y} = \mathbf{a}_\mathbf{y}, \mathbf{v}_i = \mathbf{a}_{\mathbf{v}_i} \mid x_i = a_{x_i}\big) \neq P(\mathbf{y} = \mathbf{a}_\mathbf{y},\ \mathbf{v}_i = \mathbf{a}_{\mathbf{v}_i}). \tag{17}$$

A feature x_i is *strongly relevant* if there exists some value of that feature a_{x_i}, a predictor output value $\mathbf{a}_\mathbf{y}$ and a value $\mathbf{a}_{\mathbf{v}_i}$ of a vector $\mathbf{v}_i$ for which $P(x_i = a_{x_i}, \mathbf{v}_i = \mathbf{a}_{\mathbf{v}_i}) > 0$ such that

$$P\big(\mathbf{y} = \mathbf{a}_\mathbf{y} \mid \mathbf{v}_i = \mathbf{a}_{\mathbf{v}_i}, x_i = a_{x_i}\big) \neq P(\mathbf{y} = \mathbf{a}_\mathbf{y} \mid \mathbf{v}_i = \mathbf{a}_{\mathbf{v}_i}). \tag{18}$$

Strong relevance implies that a feature is indispensable.

A feature x_i is *weakly relevant* if it is not strongly relevant, and there exists some subset of features (forming a vector $\mathbf{z}_i$) from the set of the features forming a pattern $\mathbf{v}_i$, for which there exist: some value of that feature a_{x_i}, a predictor output value $\mathbf{a}_\mathbf{y}$, and a value $\mathbf{a}_{\mathbf{z}_i}$ of a vector $\mathbf{z}_i$, for which $P(x_i = a_{x_i},\ \mathbf{z}_i = \mathbf{a}_{\mathbf{z}_i}) > 0$ such that

$$P\big(\mathbf{y} = \mathbf{a}_\mathbf{y} \mid \mathbf{z}_i = \mathbf{a}_{\mathbf{z}_i}, x_i = a_{x_i}\big) \neq P\big(\mathbf{y} = \mathbf{a}_\mathbf{y} \mid \mathbf{z}_i = \mathbf{a}_{\mathbf{z}_i}\big). \tag{19}$$

A weak relevance indicates that a feature might be dispensable.

A feature is *relevant* if it is either *strongly relevant* or *weakly relevant*, otherwise it is *irrelevant*. We can see that irrelevant features can be removed. Rough sets (Pawlak, 1991; Skowron, 1990) define strong and weak relevance for discrete features and discrete targets. For a given data set, the set of all strongly relevant features forms a *core*. A minimal set of features satisfactory to describe concepts in a given data set, including a core and possibly some weakly relevant features, forms a *reduct*. As we remember, the core is an intersection of reducts.

3.2. Methods of Feature Selection

Feature selection methods contain two main streams (Bishop, 1995; Duda and Hart, 1973; Fukunaga, 1990; John *et al.*, 1994; Pregenzer, 1997): open-loop methods and closed-loop methods.

The *open-loop* methods (*filter methods*) are based mostly on selection of features using the between-class separability criterion (Cios *et al.*, 1998; Duda and Hart, 1973). The *closed-loop* methods (John *et al.*, 1994) called also the *wrapper methods*, are based on feature selection using a predictor performance (and thus forming a 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 the pattern's elements.

3.3. Feature Selection Criteria

A feature selection algorithm is based on the defined feature selection criterion. Some of the criteria might satisfy the *monotonicity* property

$$J_{\text{feature}}(X^{+}_{\text{feature}}) \geq J_{\text{feature}}(X_{\text{feature}}), \tag{20}$$

where X_{feature} describes a feature subset, and X^{+}_{feature} denotes a larger feature subset containing X_{feature} as a subset. This means that adding a feature to a given feature set will cause the value of the criterion stay the same or increase:

$$\begin{aligned} J_{\text{feature}}\big(\{x_1\}\big) \;&\leq\; J_{\text{feature}}\big(\{x_1, x_2\}\big) \leq J_{\text{feature}}\big(\{x_1, x_2, x_3\}\big) \\ &\leq \cdots \leq J_{\text{feature}}\big(\{x_1, x_2, \ldots, x_n\}\big). \end{aligned} \tag{21}$$

Criteria with monotonicity properties cannot be used to compare the goodness of different size feature subsets when a large subset contains a smaller one. However, in practice, these criteria can still be used to compare different feature subsets of equal size.

3.3.1. Open-Loop Feature Selection Criteria

Open-loop feature selection criteria are based on information (like interclass separability) contained in the data set alone. They do not use a feedback from the predictor quality for the feature selection process.

Some of the criteria for feature selection which are based on interclass separability have the roots in the idea of Fisher's linear transformation. According to this idea, a good feature (with a high discernibility power) will cause a small within-class scatter and a large between-class scatter.

Let us consider the original (total) data set T_{all} containing N_{all} cases $(\mathbf{x}^i, c^i_{\text{target}})$ with patterns $\mathbf{x}$ constituted using n-features and labeled by one target class c^i_{target} from all l possible classes. For a data set T_{all} we will denote the number of cases in each class c_i $(i = 1, 2, \ldots, l)$ by N_i $(\sum_{i=1}^{l} N_i = N_{\text{total}})$. In order to define the feature selection criterion one needs to define a function which gives a larger value when a within-class scatter is smaller or a between-class scatter is larger (Duda and Hart, 1973; Fisher, 1936). The following criterion, based on interclass separability, may be defined:

$$J_{\text{feature}} = \frac{|\mathbf{S}_b|}{|\mathbf{S}_w|} = \frac{\det(\mathbf{S}_b)}{\det(\mathbf{S}_w)}, \tag{22}$$

$$\mathbf{S}_w = \sum_{i=1}^{l} \sum_{j=1, \mathbf{x}^j \in c_i}^{N_i} \left(\mathbf{x}^j - \boldsymbol{\mu}_i\right)\left(\mathbf{x}^j - \boldsymbol{\mu}_i\right)^T, \quad \mathbf{S}_b = \sum_{i=1}^{l} N_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \tag{23}$$

where $\boldsymbol{\mu}$ represents the *total data mean* and the determinant $|\mathbf{S}_b|$ denotes a scalar representation of the between-class scatter matrix, and similarly, the determinant $|\mathbf{S}_w|$ denotes a scalar representation of the within-class scatter matrix.

Criteria based on minimum concept description. Based on the minimum construction idea (Blumer *et al.*, 1987) and the minimum description length (Rissanen, 1978) paradigm, one technique of best feature selection could be to choose a minimal feature subset that fully describes all the concepts (e.g. classes in prediction-classification) in a given data set (Almuallim and Dietterich, 1991; Pawlak, 1991).

A straightforward technique of best feature selection could be choosing a minimal feature subset that fully describes all the concepts (for example, classes in classification) in a given data set (Almuallim and Dietterich, 1991; Doak, 1992; Kononenko, 1994; Pawlak, 1991). The idea of feature selection, with the minimum concept criterion, can be extended by using the concept of reduct defined by the theory of rough sets (Pawlak, 1991; Skowron, 1990). A reduct is a minimal set of attributes that describes all the concepts in a data set. A data set may have many reducts. If we use the definition of the above open-loop feature selection criterion, we will see that for each reduct (defining a subset of attributes $X_{\text{feature,reduct}}$) we have a maximum value of the criterion $J_{\text{feature}}(X_{\text{feature,reduct}})$. Based on the paradigm of the minimum concept description, we can select a minimum length reduct as the best feature subset.

3.3.2. Closed-Loop Feature Selection Criteria

We will consider the problem of defining a feature selection criterion for a prediction task based on an original data set T_{all} containing N_{all} cases $(\mathbf{x}, \mathbf{target})$ formed by n-dimensional input patterns $\mathbf{x}$ (whose elements represent all features X) and **targets** of the output. Assume that the m-feature subset $X_{\text{feature}} \subseteq X$ ought to be

evaluated based on the closed-loop type criterion. A reduced data set T_{feature}, with patterns containing only m features from the subset X_{feature}, should be constructed. Then a type of predictor PR_{feature} (for example, k-nearest neighbors, or a neural network), used for feature quality evaluation, should be decided. Ideally, this predictor should be the same as a final predictor PR for the whole design. However, in a simplified suboptimal solution, a computationally less expensive predictor can be used only for the feature selection purpose. Let us assume that, for the feature set X considered, a reduced feature data set X_{feature} has been selected and a predictor algorithm PR_{feature}, used for feature evaluation, had been decided. Then, evaluation of feature goodness can be provided by means of 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 which will show how to estimate the performance through the averaging of results. Consider as an example a hold-out error counting method for predictor performance evaluation. In order to evaluate the performance of the predictor PR_{feature}, an extracted feature data set T_{feature} is split into an N_{tra} case training set $T_{\text{feature,tra}}$, and an N_{test} case test set $T_{\text{feature,test}}$ (hold out for testing). Each case $(\mathbf{x}_f^i, \mathbf{target}^i)$ of both sets contains a feature pattern $\mathbf{x}_f$ labeled by a **target**. The evaluation criteria can be defined separately for prediction-classification and prediction-regression.

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

$$J_{PR_{\text{feature}}} = \hat{J}_{\text{all miscl}} = \frac{n_{\text{all miscl}}}{N_{\text{test}}} 100\%, \tag{24}$$

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 an error (expressed in percents) by the relative frequency of an error.

3.4. Feature Selection with Individual Feature Ranking

One of the straightforward feature selection procedures is based on an evaluation of the predictive power of individual features, followed by a ranking of such evaluated features and eventually the choice of the first best m features. A criterion applied to an individual feature could be either of the open-loop or closed-loop type. It can be expected that a single feature alone may have a very low predictive power, whereas when put together with others, it may demonstrate a significant predictive power.

One can attempt to select a minimal number $\hat{m}$ of the best ranked features that guarantees a performance better than or equal to a defined level according to a certain criterion $J_{\text{feature,ranked}}$.

An example of the algorithm for feature selection with individual feature ranking can be described as follows.

We assume that a data set T_{all} with N_{all} labeled patterns formed on the basis of n features $X = \{x_1, x_2, \ldots, x_n\}$ is given. We also assume that two criteria are defined: (a) an individual feature evaluation criterion $J_{\text{feature,single}}$, and (b) an evaluation criterion $J_{\text{feature,ranked}}$ for a final collection of m ranked features.

1. Set $j = 1$, and choose feature x_j.
2. Compute a predictive power of the feature x_j alone by computing the value of $J_{\text{feature,single}}(x_j)$.
3. If $j \leq n$ continue from step 1 with incremented j, $j = j + 1$, otherwise go to the next step.
4. Rank all n features according to the value of the computed criterion $J_{\text{feature,single}}$:

$$x_a, x_b, \ldots, x_m, \ldots, x_r, \quad J_{\text{feature,single}}(x_a) \geq J_{\text{feature,single}}(x_b), \text{ etc.} \tag{25}$$

5. Find the minimal number of $\hat{m}$ first ranked features according to the criterion $J_{\text{feature,ranked}}$.
6. Select the first $\hat{m}$ best ranked features as a final subset of selected features.

One of the criteria evaluating the predictive power of a feature could be defined by the *measure of significance* of the feature (attribute) $x_j \in X$,

$$\mu_{X,X}(x_j) = \frac{\operatorname{card}(POS_X(X)) - \operatorname{card}(POS_{X-\{x_j\}}(X))}{\operatorname{card} T_{\text{all}}}, \tag{26}$$

evaluated for the original classification X^* generated for the entire feature set X for data set T_{all}.

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

We will discuss PCA for feature projection and reduction, and then the joint method of feature selection using PCA and the rough sets method (Cios *et al.*, 1998).

We assume that the knowledge about a domain of recognition is represented by a limited size sample of N random n-dimensional patterns $\mathbf{x} \in \mathbb{R}^n$ representing extracted object's 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 statistically characterized by the $n \times n$ dimensional *covariance* matrix $\mathbf{R}_x$. Let the eigenvalues of the covariance matrix $\mathbf{R}_x$ be arranged in the decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq 0$ (with $\lambda_1 = \lambda_{\max}$), with the corresponding orthonormal eigenvectors $\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^n$. The optimal transformation

$$\mathbf{y} = \hat{\mathbf{W}}\mathbf{x} \tag{27}$$

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

$$\hat{\mathbf{W}} = \left[\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^m\right]^T. \tag{28}$$

This matrix is composed of m rows representing 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 $\mathbf{y}$

$$\mathbf{Y} = \left(\hat{\mathbf{W}}\mathbf{X}^T\right)^T = \mathbf{X}\hat{\mathbf{W}}^T \tag{29}$$

minimizing the mean-least-square reconstruction error. The PCA method can be applied for the 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 $\mathbf{x}$. There remains an unsolved problem of which principal components are best for a given processing goal. One of possible methods (criteria) for selection of 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.

We have applied Principal Component Analysis and the rough sets method (Cios *et al.*, 1998) for the orthonormal projection (and reduction) of reduced feature patterns representing recognized face images. In the next section, we will present an application of rough sets for feature selection/reduction.

4.1. Application of Rough Sets and Principal Components for Feature Selection

The PCA provides feature projection and reduction optimal from the point of view of minimizing the reconstruction error. However, PCA does not guarantee that the selected first principal components will be the most adequate for classification. One of the possibilities for selecting discriminative features from principal components is to apply rough sets theory (Pawlak, 1991; Skowron, 1990). Namely, a reduct can be used for selecting some of the principal components that constitute this reduct. Consequently, these principal components will describe all the concepts in a data set. Suboptimal solutions can be found by choosing a minimal length reduct or a dynamic reduct as the selected set of principal components forming a selected, final feature pattern.

The following steps can be proposed for the PCA and rough sets-based procedure for feature selection. Rough sets methods require that a processed data set contain discrete features, so the projected PCA pattern features must be discretized.

Assume that we are given a data set T, containing N cases labeled by the associated classes

$$T = \left\{(\mathbf{x}^1, c^1_{\text{target}})(\mathbf{x}^2, c^2_{\text{target}}), \ldots, (\mathbf{x}^N, c^N_{\text{target}})\right\}. \tag{30}$$

A case $(\mathbf{x}^i, c^i_{\text{target}})$ $(i = 1, 2, \ldots, N)$ is constituted with an n-dimensional real-valued pattern $\mathbf{x} \in \mathbb{R}^n$ with the corresponding categorical target class c^i_{target}. We assume that 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

$$\mathbf{X} = \left[\mathbf{x}^1,\ \mathbf{x}^2, \ldots, \mathbf{x}^N\right]. \tag{31}$$

The PCA procedure is applied for the pattern matrix $\mathbf{X}$, with the resulting $n \times n$ optimal Karhunen-Loéve matrix $\mathbf{W}_{KL}$ (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 $\mathbf{W}_{KL}$, containing only first m rows of a full size matrix $\mathbf{W}$, is constructed. Applying the matrix $\mathbf{W}_{KL}$, the original n-dimensional pattern $\mathbf{x}$ can be projected using transformation $\mathbf{y} = \mathbf{W}_{KL}\mathbf{x}$ onto the reduced m-dimensional patterns $\mathbf{y}$ in the principal components space. The entire projected $N \times m$ matrix $\mathbf{Y}$ of patterns can be obtained by the formula $\mathbf{Y} = \mathbf{X}\mathbf{W}^T_{KL}$.

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 the corresponding target classes from the original data set T. It forms a decision table DT_m with m-dimensional principal component related patterns. From the decision table DT_m, the selected reduct $X_{\text{feature,reduct}}$ of size r can be found as a final selected attribute set.

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 these columns from the discrete matrix $\mathbf{Y}_d$ which are included in the selected feature set $X_{\text{feature,reduct}}$. Each pattern in $DT_{f,d}$ is labeled by the corresponding target class. Similarly, one can obtain a real-valued reduced decision table $DT_{f,r}$ extracting (and adequately labeling by classes) these columns from the real-valued projected matrix $\mathbf{Y}$ which are included in the selected feature set $X_{\text{feature,reduct}}$. Both the resulting reduced decision tables can be used for the 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 $\{(\mathbf{x}^1, c^1_{\text{target}}), (\mathbf{x}^2, c^2_{\text{target}}), \ldots, (\mathbf{x}^N, c^N_{\text{target}})\}$.

1. Extract from the original class-labeled data set T a pattern part as an $N \times n$ data pattern matrix $\mathbf{X}$.

2. For the matrix $\mathbf{X}$ compute the covariance matrix $\mathbf{R}_x$.

3. For the matrix $\mathbf{R}_x$ find the eigenvalues and corresponding eigenvectors, and arrange them in a descending order.

4. Select the reduced dimension $m \leq n$ of a feature vector in the principal components space using the defined selection method, which may base on a judgement of the ordered values of computed eigenvalues.

5. Compute the optimal $m \times n$ Karhunen-Loéve transform matrix $\mathbf{W}_{KL}$ based on the eigenvectors of $\mathbf{R}_x$.

6. Transform the original patterns from $\mathbf{X}$ into m-dimensional feature vectors in the principal component space by formula $\mathbf{y} = \mathbf{W}_{KL}\mathbf{x}$ for a single pattern, or formula $\mathbf{Y} = \mathbf{X}\mathbf{W}_{KL}$ 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. Form the decision table DT_m using the patterns from the matrix $\mathbf{Y}_d$ with the corresponding classes from the original data set T.

9. Find a selected reduct from the decision table DT_m treated as a selected set of features $X_{\text{feature,reduct}}$ describing all the concepts in DT_m.

10. Construct the final (reduced) discrete-attribute decision table $DT_{f,d}$ containing these columns from the projected discrete matrix $\mathbf{Y}_d$ which correspond to the selected feature set $X_{\text{feature,reduct}}$. Label patterns by the corresponding classes from the original data set T.

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

The results of the discussed method of feature extraction/selection depend on the data set type and the designer decisions, including: (a) selection of the dimension $m \leq n$ of the projected pattern in the principal component space, (b) the discretization method, and (c) selection of a reduct.

5. Numerical Experiments for Face Recognition

To demonstrate the role of rough sets methods for feature selection/reduction, we have carried out numerical experiments regarding face recognition. Feature extraction from images was provided by Singular Value Decomposition. Each gray-scale face image was of the dimension 112×92 pixels. Classification of face images was performed with a Learning Vector Quantization (LVQ) neural network.

5.1. Singular Value Decomposition as a Feature Extraction from Face Images

Singular Value Decomposition (SVD) can be used to extract features from images (Hong, 1991; Świniarski and Hargis, 2001). 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 means of SVD. Assume that the rank of $\mathbf{A}$ is $r \leq m$. The matrices $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ are non-negative, symmetric, and have the identical eigenvalues λ_i. For $m \leq n$ there are at most $r \leq m$ non-zero eigenvalues. The SVD transform decomposes matrix $\mathbf{A}$ into the product of two orthogonal matrices: $\boldsymbol{\Psi}$ of dimension $n \times r$ and $\boldsymbol{\Phi}$ of dimension $m \times r$, as well as a diagonal matrix $\boldsymbol{\Lambda}^{\mathbf{1/2}}$ of dimension $r \times r$. The SVD of a matrix (image) $\mathbf{A}$ is given by

$$\mathbf{A} = \boldsymbol{\Psi}\, \boldsymbol{\Lambda}^{1/2}\, \boldsymbol{\Phi}^T = \sum_{i=1}^{r} \sqrt{\lambda_i}\, \boldsymbol{\psi}_i\, \boldsymbol{\phi}_i^T, \tag{32}$$

where the matrices $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ have r orthogonal columns $\boldsymbol{\psi}_i \in \mathbb{R}^n$ and $\boldsymbol{\phi}_i \in \mathbb{R}^m$ $(i = 1, \ldots, r)$, respectively (representing orthogonal eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$). The square matrix $\boldsymbol{\Lambda}^{1/2}$ has the diagonal entries defined by

$$\boldsymbol{\Lambda}^{1/2} = \text{diag}\left(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_r}\right), \tag{33}$$

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

$$\boldsymbol{\Lambda}^{1/2} = \boldsymbol{\Psi}^T \mathbf{A}\, \boldsymbol{\Phi}. \tag{34}$$

If the 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 $\boldsymbol{\Lambda}^{\,1/2}$ can be considered as extracted features of the image. These r singular values can be arranged as an image feature vector (SVD pattern) $\mathbf{x}_{\text{svd}} = [\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_r}]^T$ of an image.

Despite the expressive power of the SVD transformation (Hong, 1991), it is difficult to say arbitrarily how powerful the SVD features could be for a classification of face images.

The r-element SVD patterns can be heuristically reduced by removing its r_r trailing elements whose values are below a 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. Data Sets

We have analyzed of 13 selected classes of face images (13 persons), with 27 instances for each class, from an Olivetti ORL face data base[1] (Samaria and Harter, 1994). Each gray-scale face image was of the dimension 112×92 pixels (with an original face space representation of 10304 pixel-based features). The entire 351-image data set, consisting 13 classes of faces with 27 instances per class, was divided into training

[1] ORL database is available at `www.cam-orl.co.uk/facedatabase.html`

and test sets: 313 cases of these images were used for the training set, and 38 final cases for the test set. Given an original face images set, we applied feature extraction using SVD of matrices representing image pixels. As a result, we obtained for each image the 92-element $\mathbf{x}_{\text{svd}}$ pattern with features being the singular values of a face matrix (arranged in a descending order). In the next processing step a heuristic reduction of SVD patterns was provided, yielding 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 reduced SVD patterns from the training set. The projected 60-element PCA patterns were then heuristically reduced to 20-element reduced PCA patterns $\mathbf{x}_{\text{svd},r,\text{pca},r}$. In the last preprocessing step the rough sets method was used for the final feature selection/reduction of the reduced PCA continuous-valued patterns. For discretization of the continuous reduced PCA features, we applied the dividing of each attribute values range into 10 evenly spaced zones. The discretized training set was used to find the six-element reduct (Cios *et al.*, 1998). This reduct was used to form the final pattern $\mathbf{x}_{\text{svd},r,\text{pca},r,\text{rs},r}$ The training and test sets (decision tables) with real-valued pattern attributes were reduced according to the selected reduct.

5.3. Learning Vector Quantization (LVQ) Neural Network Classifier

We applied an LVQ neural network for face classification using, reduced by rough sets, training and test sets. The LVQ vector quantization neural network is a static, feedforward, neuromorphic system whose weight values can be determined using a supervised learning. For a given training set $TR = \{\mathbf{x}_i, C_{x_i}\}_{i=1}^{l}$ containing l labeled n-dimensional pattern vectors $\mathbf{x} \in \mathbb{R}^n$, the network could be formed of two layers of neurons: input and output layers. The number of neurons in the input layer equals the dimension of the input pattern vectors n. The weightless neurons of the input layer just receive the input pattern element values. The output layer contains M_q neurons, where M_q is equal to the number of code-book reference vectors. The M_q neurons of the output layer are divided into M classes $\{C_i\}_{i=1}^{M}$, defined in the training set TR. Each output neuron belongs to a certain class. Few neurons may be assigned to the same class.

The neurons of the first layer are fully connected with the output layer neurons via weights. The network outputs are modeled by $\mathbf{xs} = \mathbf{W}\mathbf{x}$; $\mathbf{y} = \mathbf{F}(\mathbf{xs})$, where $\mathbf{W}$ is the weight matrix, and $\mathbf{F}$ is the output activation vector function. The purpose of the LVQ neural network is to quantize the input patterns through representing them by M_q reference vectors. These reference vectors approximate the input patterns. The learning of the LVQ network guarantees mapping of input patterns from the input pattern space $\mathbb{R}^n$ into one of the reference vectors from a limited-size code-book $W_c = \{(i,\ \mathbf{w}_i)\}_{i=1}^{M_q}$. In the LVQ learning algorithm, usually several reference vectors of the code-book are assigned to each class C_i from the training set

$$W_{C_i} = \{\mathbf{w}_j\} \quad \text{for all} \quad j \quad \text{and} \quad \mathbf{w}_j \quad \text{representing class} \quad C_i. \tag{35}$$

Kohonen (1990) proposed the following supervised learning algorithm that approximately minimizes misclassification errors of vector quantization stated as the nearest-neighborhood classification.

During supervised learning with punish-reward idea of weights adjustment, the optimal reference vectors $\mathbf{w}_i$ $(i = 1, 2, \ldots, M_q)$ of the code-book can be found as the asymptotic values of the following learning process. First, for a given input pattern $\mathbf{x}$ belonging to the class C_l, and previous values $\{\mathbf{w}_j^k\}_{i=1}^{M_q}$, the code-book reference vector which is the nearest to the vector $\mathbf{x}$ is selected:

$$j\text{-th nearest reference vector} \quad \mathbf{w}_j \quad \min_{i=1,2,\ldots,M_q} \|\mathbf{x} - \mathbf{w}_i\|. \tag{36}$$

This reference vector belongs to a certain class C_r. Then only this j-th reference vector $\mathbf{w}_j$, nearest to $\mathbf{x}$, will be adjusted in the following way:

$$\begin{aligned} \mathbf{w}_j^{k+1} &= \mathbf{w}_j^k + \alpha(k)[\mathbf{x} - \mathbf{w}_j] \quad \text{if} \quad C_l = C_r, \\ \mathbf{w}_j^{k+1} &= \mathbf{w}_j^k - \alpha(k)[\mathbf{x} - \mathbf{w}_j] \quad \text{if} \quad C_l \neq C_r, \\ \mathbf{w}_i^{k+1} &= \mathbf{w}_i^k \quad \text{if} \quad i \neq j, \end{aligned} \tag{37}$$

where $0 < \alpha(k) < 1$ is the learning rate (a decreasing function of the learning step). The above weight adjustment is based on the "Winner-Takes-All" and punish-reward ideas. Only a reference vector $\mathbf{w}_j$, which is the nearest to the pattern $\mathbf{x}$, is adjusted.

Results of experiments. The described sequence of processing steps, applied in the design of classifiers, included: extraction of SVD features from images, heuristic reduction of SVD features, Principal Component Analysis with Karhunen-Loéve transformation, heuristic reduction of PCA features, discretization and the rough sets based feature selection and reduction. Classification of face images was performed using an LVQ neural network trained for the reduced 5-element pattern data sets. The reduction of the PCA patterns by rough sets was provided based on a selected 5-element relative reduct (containing a set of elements $\{0, 1, 2, 3, 5\}$ of the PCA pattern).

The LVQ network consisted of 5 inputs and the number of outputs dependant on the number of code-books selected to represent classes. For the reference data, the reduced training sets, the best recognition accuracy 97.3% for the test set, consisting of 38 cases, was obtained for 65 code-books, with 150,000 training epochs.

6. Conclusion

We have presented a rough sets method and its role in feature selection for pattern recognition. We proposed a sequence of data mining steps, including application of SVD, PCA and rough sets for feature selection. This processing sequence was shown as potential for feature extraction and feature selection in designing neural network classifiers for face images. This method provides a substantial reduction of the pattern dimensionality. Rough sets methods showed an ability to reduce significantly the pattern dimensionality and proved to be data mining techniques viable as a front end of neural network classifiers. The Learning Vector Quantization neural network was found as a viable classifier for patterns reduced by the rough sets method and representing facial images, yielding 97.3% of the classification accuracy for the test set.

References

Almuallim H. and Dietterich T.G. (1991): *Learning with many irrelevant features.* — Proc. 9th Nat. Conf. *Artificial Intelligence*, Menlo Park, CA, AAAI Press, pp.574–552.

Atkeson C.G. (1991): *Using locally weighted regression for robot learning.* — Proc. IEEE Int. Conf. *Robotics and Automation*, pp.958–963

Bazan J., Skowron A. and Synak P. (1994a): *Market data analysis: A rough set approach.* — ICS Res. Rep., No.6, Warsaw University of Technology, Warsaw, Poland.

Bazan J., Skowron A. and Synak P. (1994b): *Dynamic reducts as a tool for extracting laws from decision tables.* — Proc. Symp. *Methodologies for Intelligent Systems*, Charlotte, NC, pp.16–19.

Bishop C.M. (1995): *Neural Networks for Pattern Recognition.* — Oxford: Oxford Press

Blumer A., Ehrenfeucht A., Haussler D. and Warmuth M.K. (1987): *Occam's razor.* — Inf. Process. Lett., Vol.24, pp.377–380.

Diamentras K.I. and Kung S.Y. (1996): *Principal Component Neural Networks. Theory and Applications.* — New York: Wiley.

Cios K., Pedrycz W. and Świniarski R.W. (1998): *Data Mining Methods in Knowledge Discovery.* — Boston/Dordrecht/London: Kluwer Academic Publishers.

Doak J. (1992): *An evaluation of feature selection methods and their application to computer security.* — Tech. Rep., No.CSE-92-18, University of California at Davis.

Duda R.O. and Hart P.E. (1973): *Pattern Recognition and Scene Analysis.* — New York: Wiley.

Fisher R.A. (1936): *The use of multiple measurements in taxonomy problems.* — Annals of Eugenics, Vol.7, pp.179–188.

Fukunaga K. (1990): *Introduction to Statistical Pattern Recognition.* — New York: Academic Press.

Geman S., Bienenstock E. and Doursat R. (1992): *Neural networks and the bias/variance dilemma.* — Neural Comput., Vol.4, No.1, pp.1–58.

Holland J.H. (1992): *Adaptation of Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* — MIT Press.

Hong Z.Q. (1991): *Algebraic Feature Extraction of Image for Recognition.* — Pattern Recognition, Vol.24, No.3, pp.211–219.

Jain A.K. (1989): *Fundamentals of Digital Image Processing.* — New York: Prentice Hall.

John G., Kohavi R. and Pfleger K. (1994): *Irrelevant features and the subset selection problem.* — Proc. 11th Int. Conf. *Machine Learning (ICML-94)*, pp.121–129.

Karhunen K. (1947): *Uber lineare methoden in der Wahrscheinlichkeitsrechnung.* — Annales Acedemiae Scientiarum Fennicae, Series AI: Mathematica-Physica, 3rd Ed.: Van Nostrand, pp.373–379.

Kira K. and Rendell L.A. (1992): *A practical approach to feature selection.* — Proc. 9th Int. Workshop *Machine Learning*, Aberdeen, Scotland, pp.259–256.

Kittler J. (1986): *Feature selection and extraction*, In: Handbook of Pattern Recognition and Image Processing (T.Y. Young and K.S. Fu, Eds.), San Diego: Academic Press, pp.59–83.

Kohonen T. (1990): *The Self-Organizing Map.* — Proc. IEEE, Vol.78, pp.1464–1480.

Kononenko I. (1994): *Estimating attributes: Analysis and extension of Relief.* — Proc. Europ. Conf. *Machine Learning*.

Langley P. and Sage S. (1994): *Selection of relevant features in machine learning.* — Proc. AAAI Fall Symp. *Relevance*, pp.140–144.

Lewler E.L. and Wood D.E. (1966): *Branch and bound methods: A survey.* — Oper. Res., Vol.149, pp.4.

Liu H. and Setiono R. (1996a): *A probabilistic approach to feature selection—A filter solution.* — Proc. 13th Int. Conf. *Machine Learning (ICML'96)*, Bari, Italy, pp.319–327.

Liu H. and Setiono R. (1996b): *Feature selection and classification—A probabilistic wrapper approach.* — 9th Int. Conf. *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-AIE'96)*, Fukuoka, Japan, pp.419–424.

Liu H. and Motoda H. (1999): *Feature Selection for Knowledge Discovery and Data Mining.* — Dordrecht: Kluwer Academic Publishers.

Lobo V., Moura-Pires F. and Świniarski R. (1997): *Minimizing the number of neurons for a SOM-based classification, using Boolean function formalization.* — Int. Rep., San Diego State University, Department of Mathematical and Computer Sciences.

Marill T. and Green D.M. (1963): *On the effectiveness of receptors in recognition systems.* — IEEE Trans. Inf. Theory, Vol.9, pp.11–17.

Modrzejewski M. (1993): *Feature selection using rough sets theory.* — Proc. European Conf. *Machine Learning*, pp.213–226.

Narendra P.M. and Fukunaga K. (1977): *A branch and bound algorithm for feature subset selection.* — Trans. IEEE. Computers, Vol.C-26, pp.917–922.

Nguyen T. *et al.* (1994): *Application of rough sets, neural networks and maximum likelihood for texture classification based on singular value decomposition.* — Proc. Int. Workshop *RSSC Rough Sets and Soft Computing*, San Jose, U.S.A., pp.332-339.

Pal S.K. and Skowron A. (1999): *Rough-Fuzzy Hybridization: A New Trend in Decision Making.* — Singapore: Springer.

Pawlak Z. (1982): *Rough sets.* — Int. J. Comp. Sci., Vol.11, pp.341–356.

Pawlak Z. (1991): *Rough Sets. Theoretical Aspects of Reasoning About Data.* — Boston: Kluwer Academic Publishers.

Pregenzer M. (1997): *Distinction sensitive learning vector quantization.* — Ph.D. Thesis, Graz University of Technology, Graz, Austria.

Quinlan J.R. (1993): *C4.5: Programs for Machine Learning.* — New York: Morgan Kaufman.

Rissanen J. (1978): *Modeling by shortest data description.* — Automatica, Vol.14, pp.465–471.

Samaria F. and Harter A. (1994): *Parametrization of stochastic model for human face identification.* — Proc. IEEE Workshop *Application of Computer Vision*.

Siedlecki W. and Sklanski J. (1988): *On automatic feature selection.* — Int. J. Pattern Recogn. Artif. Intell., Vol.2, No.2, pp.197–220.

Skowron A. (1990): *The rough sets theory and evidence theory.* — Fundamenta Informaticae, Vol.13, pp.245–262.

Swets D.L and Weng J.J. (1996): *Using discriminant eigenfeatures for image retrieval.* — IEEE Trans. Pattern Recogn. Mach. Intell., Vol.10, No.8, pp.831–836.

Świniarski R. (1993): *Introduction to rough sets*, In: Materials of the Int. Short Course *Neural Networks. Fuzzy and Rough Systems. Theory and Applications.* — San Diego State University, San Diego, California, pp.1–24.

Świniarski R. (1995): *RoughFuzzyLab.* — A software package developed at San Diego State University, San Diego, California.

Świniarski R. and Nguyen J. (1996): *Rough sets expert system for texture classification based on 2D spectral features.* — Proc. 3rd Biennial European Joint Conf. *Engineering Systems Design and Analysis ESDA'96*, Montpellier, France, pp.3–8.

Świniarski R., Hunt F., Chalret D. and Pearson D. (1995): *Feature selection using rough sets and hidden layer expansion for rupture prediction in a highly automated production system.* — Proc. 12th Int. Conf. *Systems Science*, Wrocław, Poland.

Świniarski R. and Hargis L. (2001): *Rough sets as a front and of neural networks texture classifiers.* — Neurocomputing, Vol.36, pp.85–102.

Swingler K. (1996): *Applying Neural Networks.* — London: Academic Press.

Weiss S. and Indurkhya N. (1977): *Predictive Data-Mining: A Practical Guide.* — New York: Morgan Kaufmann.

Yu B. and Yuan B. (1993): *A more efficient branch and bound algorithm for feature selection.* — Pattern Recognition, Vol.26, No.6, pp.883–889.

Xu L., Yan P. and Chang T. (1989): *Best first strategy for feature selection.* — Proc. 9th Int. Conf. *Pattern Recognition*, pp.706–708.