# Studying System Properties with Rough Sets

Alicja Mieszkowicz-Rolka and Leszek Rolka

Department of Avionics and Control,
Rzeszów University of Technology,
ul. W. Pola 2, 35-959 Rzeszów, Poland
{alicjamr, leszekr}@prz.rzeszow.pl

**Abstract.** This paper considers the analysis of decision system properties basing on the rough sets theory. Investigation of human operator's decision model in dynamic processes is discussed. Extensions of the basic rough sets theory are presented, which are useful in case of analysing inconsistent decision tables. Rough sets aided research of data sets properties, in order to facilitate the synthesis of feedforward neural networks, is proposed. Examples based on real data are given.

## 1 Introduction

The rough sets theory was introduced by Pawlak [1,2] in early 80's years and can be used for analysing information systems expressed in form of decision tables. Many applications of the rough sets theory were proposed [3,8] and some theoretical modifications and extensions were introduced [4,6,7]. In this paper the rough sets theory will be discussed in two different contexts. The first one refers to evaluation of human operator's decision model, who controls a dynamic plant. This is contrary to the most recognised applications of the rough sets theory, which concerned industrial processes with slow dynamic or static information systems [2,8]. The second one deals with design and verification of artificial neural networks for diagnosis of the grinding wheel wear.

## 2 The Rough Sets Measures

An information system is defined as an ordered 4-tuple

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

where:
$X$ – a nonempty set, called the universe,
$Q$ – a finite set of attributes,
$V$ – a set of attributes values, $V = \bigcup_{q \in Q} V_q$,
$f$ – an information function, $f\colon\ X \times Q \to V,\ \ f(x,q) \in V_q,\ \ \forall q \in Q$ and $\forall x \in X$.

The elements of the universe $X$ can be compared by considering the values of their attributes. A binary indiscernibility (equivalence) relation $R$ defined in

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 115–124, 2003.

the universe, generates a partition of the universe into indiscernibility classes denoted by $X/R$.

Let $[x]_R$ denote an indiscernibility class that contains the element $x$.

$R$-lower approximation of a set $A \subseteq X$, denoted as $\underline{R}(A)$, is a set defined as follows:

$$\underline{R}(A) = \{x \in X\colon\ [x]_R \subseteq A\}\ . \tag{2}$$

$R$-upper approximation of a set $A \subseteq X$, denoted as $\overline{R}(A)$, is a set defined as follows:

$$\overline{R}(A) = \{x \in X\colon\ [x]_R \cap A \neq \emptyset\}\ . \tag{3}$$

The pair of sets $(\underline{R}(A), \overline{R}(A))$ is called the rough set approximating $A$.

A positive area of the set $A \subseteq X$ is a set denoted by $\mathrm{Pos}_R(A) = \underline{R}(A)$.

Each subset of attributes $P \subseteq Q$ generates in the universe $X$ a binary indiscernibility relation, denoted by $\mathrm{IND}(P)$. A decision model divides the universe into subsets (classes) denoted by $Y = \{Y_1, Y_2, \ldots, Y_n\}$. The classification $Y$ can be approximated using the indiscernibility classes obtained for a subset of attributes $P$.

$\underline{P}(Y) = \{\underline{P}(Y_1), \underline{P}(Y_2), \ldots, \underline{P}(Y_n)\}$ is a family of the lower approximations of the classification $Y$. The sum of the subsets contained in the family of lower approximations is called a positive area $\mathrm{Pos}_P(Y)$.

$\overline{P}(Y) = \{\overline{P}(Y_1), \overline{P}(Y_2), \ldots, \overline{P}(Y_n)\}$ is a family of the upper approximations of the classification $Y$.

The approximation quality of $Y$ by $P$ denoted by $\gamma_P(Y)$ is defined as follows:

$$\gamma_P(Y) = \frac{\mathrm{card}(\mathrm{Pos}_P(Y))}{\mathrm{card}(X)} = \frac{\sum_{i=1}^{n} \mathrm{card}(\underline{P}(Y_i))}{\mathrm{card}(X)} \tag{4}$$

where card denotes the cardinality of a set.

The approximation accuracy of a set $A \subseteq X$ by $P$ denoted by $\beta_P(A)$ is defined as the quotient of cardinalities of the $P$-lower approximation and the $P$-upper approximation of $A$:

$$\beta_P(A) = \frac{\mathrm{card}(\underline{P}(A))}{\mathrm{card}(\overline{P}(A))}\ . \tag{5}$$

For a classification $Y$ the definition of approximation accuracy (accuracy of classification) $\beta_P(Y)$ is given as follows:

$$\beta_P(Y) = \frac{\sum_{i=1}^{n} \mathrm{card}(\underline{P}(Y_i))}{\sum_{i=1}^{n} \mathrm{card}(\overline{P}(Y_i))}\ . \tag{6}$$

The above notation can be further expanded, when the set of attributes is divided into subsets of condition attributes $C$ and decision attributes $D$ and used to describe a system of decision rules (decision table). The classification $Y = \{Y_1, Y_2, \ldots, Y_n\}$ is then given by indiscernibility classes obtained with respect to decision attributes $D$. We can evaluate the approximation quality of the classification $Y$ by means of the indiscernibility classes generated for condition attributes $C$.

## 3 Variable Precision Rough Sets in Human Operator's Decision Model

The control actions of the operator can be expressed in form of decision table. It is necessary to specify the condition and decision attributes and take into account the lag caused by human factor and the characteristic of the controlled plant. The recorded process data are used to build a decision table, which is the base for determining the decision model of the operator. It is easy to find contradictory decisions in control of dynamic systems since the obtained universes are relatively large. The non-deterministic decision rules are always omitted (in lower approximation) in the original rough sets theory, and the calculated value of approximation quality $\gamma_C(D)$ can be low. The described problems may be avoided by using the approach of variable precision rough sets theory (VPRS) [4,6,7] which is more suitable for analysis of dynamic plant control. We proposed some modifications of VPRS.

It is possible to determine the consistence of control actions of the operator, who controls a dynamic system, and specify his decision model in form of deterministic decision rules. This is done by evaluating the measure denoted by $k_{\mathrm{AR}}$ (described bellow) that expresses the dependence of decision attributes on condition attributes.

We proposed [6] a new definition of the $\alpha$-positive area of a set $A \subseteq X$:

$$\mathrm{Pos}_R^{\alpha}(A) = \{x \in X\colon\ x \in ([x]_R \cap A) \ \text{ and } \ e([x]_R, A) \leq \alpha\} \tag{7}$$

where

$$e(A, B) = 1 - \frac{\mathrm{card}(A \cap B)}{\mathrm{card}(A)}\ . \tag{8}$$

The quantity $e(A, B)$, defined for any nonempty sets $A$ and $B$, is called the inclusion error of $A$ in $B$ [4]. The value of $\alpha$ should be limited: $0 \leq \alpha \leq 0.5$.

The notion $\mathrm{Pos}_R^{\alpha}(A)$ can be applied to a family of sets. By using it to a family of equivalence classes $D^*$ obtained from a decision table we get a new measure denoted by $k_{\mathrm{AR}}$.

The $\alpha$-approximation quality of $D^*$ by $C$ is expressed as follows:

$$k_{\mathrm{AR}} = \frac{\mathrm{card}(\mathrm{Pos}_C^{\alpha}(D^*))}{\mathrm{card}(X)} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \delta_{ij} \mathrm{card}(C_i \cap D_j)}{\mathrm{card}(X)} \tag{9}$$

where:

$$\delta_{ij} = \begin{cases} 1 & if \quad e(C_i, D_j) \le \alpha \\ 0 & \text{otherwise} \end{cases}$$

$n$ – number of equivalence classes for decision attributes $C$,
$m$ – number of equivalence classes for condition attributes $D$,
$\alpha$ – acceptable inclusion error; $0 \le \alpha \le 0.5$,
$D^* = \{D_1, D_2, \ldots, D_m\}$.

If the value of $k_{\mathrm{AR}}$ is nearly equal to 1, then the consistence (determinism) of operator's decisions is large and his decision model is constant. A low value of $k_{\mathrm{AR}}$ indicates that the decisions are random and don't depend on the values of condition attributes (assuming a complete set of condition attributes). This means that the operator's abilities to process the information are not correctly formed. One can also detect the importance of particular condition attributes and determine the subset of attributes necessary for identification of the operator's decision model. If, after removing a condition attribute, the value of $k_{\mathrm{AR}}$ is not changing, it means that the attribute is not necessary and doesn't have a significant importance in the decision process.

A lot of pilot's control protocols were analysed using the VPRS approach. The condition attributes consisted of: deviations from required values of selected flight parameters, and the changes of the former parameters as a source of prognostic information for the pilot. The decision attributes were deflection angles or changes of deflection angles of the aircraft's control elements. The values for particular condition attributes were coded as integer numbers e.g. (-3, -2, -1, 0, 1, 2, 3) basing on fixed intervals for deviations from the required values of the parameters. The boundaries of the intervals conform to valid norms or suggestions of experts. For decision attributes the linguistic values "negative", "zero", "positive" were used.

In order to analyse the control task of selected variables two methods were used and the results of them were compared:

a) method, which determine the values of $k_{\mathrm{AR}}$ for all discrete-time instants (all elements of the universe) for a given decision table,
b) method, which determine the values of $k_{\mathrm{AR}}$ for those instants only, were a qualitative change of control occurred.

**Table 1.** Analysis of decision table for altitude stabilisation

| Method | $k = \gamma_C(D)$ | $k_{\mathrm{AR}}$ $\alpha = 0.2$ | Number of elements in universe | Number of det. rules | Number of non-det. rules |
|---|---|---|---|---|---|
| a | 0.128 | 0.859 | 658 | 11 | 40 |
| b | 0.648 | 0.722 | 54 | 16 | 11 |

The examples of decision tables analysis are presented in the tables 1 and 2. The values of $k_{\mathrm{AR}}$ in the table 1 show that the consistence of operator's decisions is quite large and his decision model is rather constant. One can observe a big difference between the calculated value of approximation quality $k = \gamma_C(D)$ and the value of $k_{\mathrm{AR}}$, especially for the method a.

**Table 2.** Importance of condition attributes for altitude stabilisation (method b)

| Removed attribute | none | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| $k_{\mathrm{AR}}$ $(\alpha = 0.2)$ | 0.722 | 0.722 | 0.556 | 0.211 | 0.0 |

Basing on the evaluated importance of condition attributes we can state, in case of altitude stabilisation (table 2), that for the generated decision table the attributes: $c_2$ – rate of climb, $c_3$ – change of rate of climb and $c_4$ – previous change of rudder deflection angle are particularly important. The attribute $c_1$ – deviation of altitude from the required value has not a big importance. After analysis of many control tasks realised by pilots on a flight simulator we can observe that the VPRS modification is a useful method, which helps to identify the decision model of human operator who controls a dynamic plant.

The presented approach was combined with another (statistical) methods in order to fully describe the process of control performed by a human operator.

## 4 Application of the Rough Sets Theory to Synthesis of Artificial Neural Networks

An important question in the area of artificial neural networks is – how utilise effectively all information contained in the training data sets in the process of synthesis of feedforward neural networks. We propose to apply the rough sets measures for research of data sets expressed in form of information systems.

The rough sets measures were used during design of neural networks and applied to data sets obtained from a real classification task [5]. The problem was to build a classifier that could automatically evaluate the degree of wear of the grinding wheel, in the grinding process. This classification is commonly performed by a skilled operator (expert). In order to do the task automatically the characteristic process quantities must be measured (e.g. grinding forces, acoustic emission) and fed into the inference system. Trying to create a classifier based on neural networks was one of the used approaches.

Basing on registered process data and classification of an expert, decision tables for training and testing set were obtained. The investigations concerned classification with only two values of the decision attribute, which represent two characteristic states of the grinding wheel ($Y_1$ - sharp, $Y_2$ - blunt). An attempt was made to obtain a finer classification with one additional characteristic state.

The analysis of properties of data sets (quality of approximation, dependencies between attributes, significance of the condition attributes in the decision process) led to a conclusion, that the training patterns are suitable to create a good classifier. The results of the analysis are given in the table 3. The obtained quality of approximation for the original universal set was about 0.9, which is a good value in real applications. The properties of independently generated training and testing sets were qualitatively similar. When these properties are comparable for both the sets, a conclusion can be made, that both the training and testing sets are similarly representative for the considered problem.

**Table 3.** The approximation quality and accuracy of the training set

| Removed attributes | $\gamma_P(Y)$ | $\beta_P(Y)$ | $\beta_P(Y_1)$ | $\beta_P(Y_2)$ |
|---|---|---|---|---|
| none | 0.88 | 0.78 | 0.78 | 0.78 |
| $c_1$ | 0.82 | 0.69 | 0.68 | 0.70 |
| $c_2$ | 0.65 | 0.49 | 0.49 | 0.48 |
| $c_3$ | 0.79 | 0.65 | 0.63 | 0.66 |
| $c_4$ | 0.88 | 0.78 | 0.78 | 0.78 |
| $c_5$ | 0.78 | 0.64 | 0.64 | 0.64 |
| $c_6$ | 0.38 | 0.24 | 0.22 | 0.26 |
| $c_7$ | 0.88 | 0.78 | 0.78 | 0.78 |
| $c_5$, $c_6$ | 0.15 | 0.087 | 0.079 | 0.072 |
| $c_5$, $c_7$ | 0.77 | 0.63 | 0.63 | 0.63 |
| $c_6$, $c_7$ | 0.38 | 0.24 | 0.22 | 0.26 |
| $c_5$, $c_6$, $c_7$ | 0.074 | 0.038 | 0.046 | 0.031 |

In the next step a feedforward neural network with one hidden layer was created and trained. Various network structures (number of hidden neurones) were tested. Good training results have been received because the consistency of the training set was high (0.9).

In another case the approximation quality for training set was only about 0.61 (table 4) and we could expect poor training results of a neural classifier as shown in the table 5. It is of course possible to increase the number of hidden neurones and improve the learning on the training set, but we could easily verify (using the testing set) that such a neural classifier is not a good one.

**Table 4.** Training set with low approximation quality

| Removed attributes | $\gamma_P(Y)$ | $\beta_P(Y)$ | $\beta_P(Y_1)$ | $\beta_P(Y_2)$ |
|---|---|---|---|---|
| none | 0.61 | 0.41 | 0.46 | 0.34 |

We made also an attempt to modify the training algorithm. The importance of patterns in the training set is not equal. It is possible to divide the training set into subsets, which could be classified with various number of condition attributes (a condition attribute is identical with a network input, and a decision

**Table 5.** Training results for low approximation quality (network structure 7-4-3)

| Trial<br>number | Number<br>of present. | Training<br>error [%] |
|---|---|---|
| 1 | 9005 | 4.0 |
| 2 | 5154 | 2.8 |
| 3 | 7223 | 5.2 |
| 4 | 5602 | 2.7 |
| 5 | 5918 | 5.2 |

attribute corresponds to a network output). One can generate a division of the universe basing on minimal number of condition attributes needed to classify each of the training examples. We can use a modified method of presenting the examples to the network during training, by regarding the ability of classification for each of the training patterns. In the standard training algorithm each of the training examples is presented with an equal frequency. It seems reasonable to base the number of presentations of the training patterns on the number of condition attributes needed to classify them.

Let denote by $X_k$ a subset of elements of the training patterns universe $X$, which require for a correct classification at least $k$ most significant condition attributes. For a small value of $k$ the subset $X_k$ may be empty.

We can assign to each nonempty subset $X_k$ a coefficient $c_k$ denoting the frequency of presentation of a given subset during the training of the network. If an equal frequency for all training examples (probability for a random presentation) is applied, then the following condition is fulfilled:

$$c_k = \frac{\mathrm{card}(X_k)}{\mathrm{card}(X)} \ . \tag{10}$$

When regarding different classification ability of the subsets $X_k$, the coefficients $c_k$ can be modified. For a subset $X_k$, which contains training patterns, that require a greater number of condition attributes for classification, the coefficient $c_k$ may be slightly increased. Such a modification of the network training is equivalent to seeking for minimum of a modified cost function $E_\mathrm{m}$, in which some of the elements in the sum of the square errors would me multiplied by a number greater then 1. The modified cost function $E_\mathrm{m}$ has generally a greater value than the original cost function $E$ ($E_\mathrm{m} \geq E$). Thus, reaching the required value for the modified cost function also delivers an acceptable value for the original cost function.

The training patterns, divided into subsets based on the number of the most significant condition attributes needed for their classification, are shown in the table 6. The neural network was then trained using the subsets $X_4$ and $X_5$ more frequently. This resulted in some cases in a better convergence of the training (table 7). The removal of contradictory training patterns was important to attain a better quality of the training set.

**Table 6.** Training subsets with respect to the most significant condition attributes

| Subset | Condition attributes | Approximation quality | Number of elements |
|---|---|---|---|
| $X_5$ | 1, 2, 3, 5, 6 | 0.88 | 14 |
| $X_4$ | 2, 3, 5, 6 | 0.77 | 47 |
| $X_3$ | 2, 5, 6 | 0.43 | 5 |
| $X_2$ | 2, 6 | 0.39 | 53 |
| $X_1$ | 6 | 0.00 | 0 |

**Table 7.** Training results for different presentation of sets (network structure 7-4-2)

| Required training error [%] | Additional presentation of set | Number of trials | Total number of present. |
|---|---|---|---|
| 0.5 | - | 1 | 240 |
| 0.5 | $U_5$ | 1 | 192 |
| 0.5 | $U_4$ | 1 | 149 |
| 0.05 | - | 7 | 94163 |
| 0.05 | $U_5$ | 1 | 7697 |
| 0.05 | $U_4$ | 7 | 56422 |

In the last step the importance of connections between neurones for created variants of neural network was researched. It was an interesting stage, during which the possibility to discover superfluous connection was confirmed and effectively used. The table 8 contains the values of approximation quality for a hidden layer and for single neurones in that layer. In the table 9 the values of approximation quality for the output layer are given.

It must be emphasised that the training patterns itself were not changed in any way by the algorithms used to investigate the properties of the patterns set. A random character of the training algorithm was maintained in order to effectively reach a minimum of the cost function. The obtained practical results confirm the usefulness of the proposition given in this paper. Nevertheless more complex investigations would be necessary to make well-founded conclusions. We suggest in any case to carry out an research of data set properties when designing neural networks.

## 5 Conclusions

The rough sets theory and its extensions (e.g. VPRS) can be relatively easy applied to analysis of data sets expressed in form of decision tables. Basing on various rough sets measures one is able to discover the properties of information systems and to optimise decision rules by finding the reducts of attributes. The variable precision rough sets theory (VPRS) was especially helpful in research

**Table 8.** Approximation quality for the hidden layer (network structure 7-4-2)

| Removed input | $\gamma_P(Y)$ | $\gamma_{P1}(Y)$ | $\gamma_{P2}(Y)$ | $\gamma_{P3}(Y)$ | $\gamma_{P4}(Y)$ |
|---|---|---|---|---|---|
| none | 0.82 | 0.91 | 1.00 | 0.90 | 0.94 |
| 1 | 0.77 | 0.90 | 0.98 | 0.87 | 0.93 |
| 2 | 0.59 | 0.85 | 1.00 | 0.73 | 0.82 |
| 3 | 0.73 | 0.88 | 1.00 | 0.86 | 0.90 |
| 4 | 0.79 | 0.91 | 1.00 | 0.90 | 0.92 |
| 5 | 0.72 | 0.91 | 1.00 | 0.81 | 0.94 |
| 6 | 0.34 | 0.70 | 1.00 | 0.61 | 0.52 |
| 7 | 0.82 | 0.91 | 1.00 | 0.90 | 0.94 |

**Table 9.** Approximation quality for the output layer (network structure 7-7-2)

| Removed input | $\gamma_P(Y)$ |
|---|---|
| none | 1.00 |
| 1 | 0.74 |
| 2 | 1.00 |
| 3 | 1.00 |
| 4 | 1.00 |
| 5 | 0.82 |
| 6 | 0.60 |
| 7 | 0.36 |

of large universes that are obtained in case of control of dynamic systems. The VPRS measures are not so "restrictive" as those of the original rough sets theory and are better suited for analysis of inconsistent decision tables. The rough sets theory can be used on various stages of neural networks design to: optimise training and testing sets, asses representativeness of training and testing sets, predict convergence of training by inspecting consistency of training patterns, modify pattern presentation using the evaluated significance of training examples, analyse and optimise (reduce) trained networks. We can conclude that the rough sets theory constitutes an useful framework especially for creating hybrid soft computing systems.

## References

1. Pawlak, Z.: Information systems. Theoretical fundamentals. WNT, Warszawa (1983) (in Polish)
2. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht Boston London (1991)
3. Słowiński, R. (ed.): Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets. Kluwer Academic Publishers, Dordrecht Boston London (1992)

4. Ziarko, W.: Analysis of Uncertain Information in the Framework of Variable Precision Rough Sets Modelling. Proceedings of Workshop. Rough Sets. State of Art and Perspective. Kiekrz, Poland (1992)
5. Rolka, L., Mieszkowicz-Rolka, A.: The application of rough sets and fuzzy sets in the grinding process. Proceedings of the 4-th Conference – Knowledge Engineering and Expert Systems. Wrocław, (2000) (in Polish)
6. Mieszkowicz-Rolka, A., Rolka, L.: Variable Precision Rough Sets in Analysis of Inconsistent Decision Tables. In: Rutkowski, L., Kacprzyk, J. (eds.): Advances in Soft Computing. Proceedings of the Sixth International Conference on Neural Network and Soft Computing. Zakopane, Poland, Physica-Verlag (2003)
7. Mieszkowicz-Rolka, A., Rolka, L.: Variable Precision Rough Sets. Evaluation of Human Operator's Decision Model. In: Sołdek, J., Drobiazgiewicz, L. (eds.): Advanced Computer Systems. Proceedings of the 9-th International Conference on ACS. Międzyzdroje, Szczecin, Poland (2002)
8. Mrózek, A.: Inference Models of Human Operators and their Application in Computerization of Technological Objects. Scientific Papers of the Silesian University of Technology, Gliwice (1989) (in Polish)