On Construction of Optimised Rough Set-based Classifier

Urszula Stanczyk

***Abstract*—One of popularly used forms of definition for a classifier is a decision algorithm constructed from conditional clauses of "IF...THEN..." type. Extraction of decision rules that comprise such a decision algorithm constitutes one of crucial steps within the rough set approach to the problem of classification. The first step of the process is to find all relative raducts of conditional attributes and to select one, if several exist, for the computations that follow. The second phase is taken by the procedure of establishing all valid relative value reducts. From the variety of possible solutions there is required the one with the highest accuracy of classification as well as simplicity of implementation which is reflected by the lowest number of conditional clauses within the decision algorithm. In the paper there is described such optimising methodology employed to the rough set-based approach to the stylometric problem of authorship attribution.**

***Index Terms*—Optimisation, Covering, Stylometry, Rough sets, Authorship attribution, Relative reduct, Relative value reduct.**

I. Introduction

STYLOMETRIC methods, which belong to the category of text mining [1] or even wider to data mining, comprise textual analysis of written texts that yields information on linguistic style of their authors and these styles themselves and it can be used in academic, literary, legal or even forensic applications to detect cases of plagiarism, for texts of disputed authorship to find the real authors, to establish similarities between some writers, or define their unique characteristics called author invariants [2].

Techniques employed to stylometric analysis use computational powers of computers applying typically either statistical approaches or machine learning methodologies [3]. Within the latter group of techniques there is included the rough set theory and its elements.

Classical rough set theory (RST) is usually enumerated among other approaches that deal with imperfect or incomplete knowledge about the Universe [4], the most famous example of which constitutes probably the fuzzy set theory due to Lotfi Zadeh [5].

Rough set theory was developed by Polish scientist Zdzislaw Pawlak [6] in the early 1980s and it provides tools for interpretation and manipulation of incomplete knowledge of objects perceived in form of granules based on indiscernibility relations defined for conditional and decision attributes.

The knowledge about the Universe presented in the decision table is quite often redundant and can be expressed more compactly by exploiting the concept of relative reducts which are such subsets of attributes that preserve the classification properties of the decision table even though some of conditional attributes are disregarded [7]. Finding and selecting relative reducts for a given decision table can be time-consuming process of high computational complexity in case of large sets of attributes and there are invented various methods [8] to tackle this problem efficiently, for example by taking into account the importance of some attributes.

Further optimisation within the rough set-based approach offers the stage that follows the choice of a relative reduct, and it is generation and then selection of relative value reducts [9]. Relative value reducts can be perceived as masks put on decision rules included in the decision table, indicating for each rule these attributes whose values are sufficient to perform correct classification. It is quite common that for a decision rule several distinct relative value reducts can be used and this results in the necessity of choice among them. Thus there arises the question of criteria used in such selection and its consequences [10] for classification accuracy and optimality of execution for the constructed decision algorithm that can contain various numbers of conditional clauses. Actually all decision rules, regardless of the number of attributes they are composed of, are in fact conditional clauses, yet for the simplicity sake to distinguish the shortened rules received from relative value reducts only these will be referred to as conditional clauses in the paper.

The paper describes how the notion of coverage from optimisation-motivated algorithms [11] can be employed in considerations on possible relative value reducts and their choice within the process of the rules extraction for the decision algorithm, and how it reflects upon the number of decision clauses and the classification accuracy in the task of rough set-based approach to authorship attribution of literary texts.

II. Stylometric Tasks and Techniques

Contemporary stylometry can be seen as a successor of historical textual analysis that by cumbersome comparisons of documents have led to proving or disproving the authenticity or authorship of written texts.

Typically there are distinguished three main stylometric tasks:

- author characterisation - dedicated to finding some unique elements that define a writer's style, also some elements of the writer's background,

U. Stanczyk is with the Institute of Informatics, Silesian University of Technology, Gliwice, 44-100 Poland, phone: +48-32-237-2969; fax: +48-32--237-2733; e-mail: urszula.stanczyk@polsl.pl.

Manuscript received December 31, 2008; revised ...

- author comparison - focused on establishing, if they exist, some properties in common between texts authored by different writers,
- author attribution - that provides means for settling questions of disputed authorship.

Although writer characterisation and comparison can be considered just by themselves, writer attribution encompasses both of them and more as without unique definition of writing style that distinguishes it from others the task of finding the true author of some disputed text would be unsolvable.

In the early years of its history linguistic analysis relied on finding some distinct features of texts such as language structures or vocabulary. Yet this kind of textual descriptors cannot be generally perceived as reliable because they are too prone to forgery, and the origins of modern stylometry are usually dated to 1787 when Edmond Malone, an expert on Shakespeare's plays, in his published works argued the usage of quantitative over qualitative descriptors such as meter and rhyme, or to 1851 when Augustus de Morgan proposed the usage of average word lengths, which was made famous by study of T.C. Mendenhall on word-length distributions printed in 1887 [12].

These early attempts gave rise to one branch of typically applied stylometric methods which employ all kinds of statistics and nowadays heavily rely on computer assisted computations of probabilities and distributions of single letters and other characters, single words, word patterns, or patterns of sentences. As examples of such methodology among others there can be mentioned Markov chains ([13] and [14]), cumulative sum (QSUM or CUSUM) [15], principal component analysis (PCA), linear discriminant analysis (LDA), cluster analysis.

Another group of stylometric approaches constitute machine learning algorithms in which identification of author or author characteristics is considered as any other classification problem to be solved. Popularly there are used artificial neural networks [16], genetic algorithms [17], support vector machines, decision trees.

In this latter group of techniques there is also included rough set-based approach, presented in more detail in the next section of the paper.

Current trends in science lead to fusion of approaches if any single one does not satisfy requirements and this is often true for classifiers. Hence there are also constructed hybrid solutions, for example incorporating rough and fuzzy sets theories ([18] and [19]), or genetic algorithms and cluster analysis [17] to arrive at higher classification accuracy.

Apart from the choice of applied methodology another essential problem of stylometric analysis is the type of textual descriptors which express the knowledge about the Universe that is feature selection for intended classification [20]. Even the most efficient classification technique cannot help if the knowledge about training samples is insufficient to construct a classifier.

Typically there are used four types of textual descriptors:

- lexical - statistics such as total number of words, average number of words per sentence, distribution of word length, total number of characters (including letters, numbers and special characters such as punctuation marks), frequency of usage for individual letters, average number of characters per sentence, average number of characters per word, etc.,
- syntactic - describe such patterns of sentence construction as formed by punctuation,
- structural - reflect the general layout of text that is its organisation into headings or paragraphs and elements like font type, embedded pictures or hyperlink,
- content-specific - words of higher importance or with specific relevance to some domain.

As with techniques, also with text markers there can be used hybrid approaches involving for example both lexical and syntactic descriptors.

III. Rough Set-Based Classification

The first step in the rough set-based approach to classification problem is defining a Decision Table that contains the whole knowledge about the Universe of discourse (U). Columns of the Decision Table are defined by conditional (C) and decision (D) attributes while rows (X) specify values of these attributes ($A = C \bigcup D$) for each object of the Universe, which allow to partition U into equivalence classes ($[x]_A$) basing on the notion of indiscernibility relation [6].

The indiscernibility relation and resulting from it equivalence classes enable to describe sets of objects by their lower $\underline{A}X$ and upper approximations $\overline{A}X$. In the lower approximation there are included these objects of the Universe for which the entire equivalence class also is included in the considered set, while the upper approximation is constructed with these objects for which at least one element of the equivalence class is included in the set. Set difference between the upper and lower approximation being empty indicates that the set is crisp, otherwise it is said to be rough.

The Decision Table (DT) is defined as 5-tuple

$$DT =< U, C, D, v, f > \tag{1}$$

U, C, and D being finite sets, and v such a mapping that to every $a \in C \bigcup D$ assigns its finite value set V_a (domain of attribute a), and f the information function $f : U \times (C \bigcup D) \rightarrow V$, with V being the union of all V_a and $f(x, a) = f_x(a) \in V$ for all x and a.

Information held by a Decision Table is often excessive in such sense that either not all attributes or not all their values are needed for correct classification of objects. For such occasions within the rough set approach there are included dedicated tools that enable to find, if they exist, such functional dependencies between attributes that allow for decreasing their number without any loss of classification properties of DT, and these are relative reducts and relative value reducts.

A relative reduct of attributes C with respect to D, $RED_D(C)$, is defined as the maximum independent subset of attributes $R \subseteq C$. For C-positive region of the family D^*, $POS_C(D^*)$, defined as

$$POS_C(D^*) = \bigcup_{X_i \in D^*} \underline{C} D_i \tag{2}$$

if R is D-reduct, then $POS_R(D^*) = POS_C(D^*)$ and $C \xrightarrow{k} D$ implicates $R \xrightarrow{k} D$.

An attribute $c \in C$ is said to be redundant in C with respect to D when

$$POS_C(D^*) = POS_{C-\{c\}}(D^*) \tag{3}$$

otherwise it is irremovable from C with respect to D.

The set of all D-irremovable attributes of C constitutes a relative core of C with respect to D

$$CORE_D(C) = \{c \in C: \\ POS_C(D^*) \neq POS_{C-\{c\}}(D^*)\} \tag{4}$$

D-reduct and D-core are in relation

$$CORE_D(C) = \bigcap_{R \in RED_D(C)} R \tag{5}$$

It is possible that for one Decision Table several relative reducts exist, especially when the number of conditional attributes is high, thus the process of finding all of them and the final choice can be quite time-consuming. Since all relative reducts preserve classification properties of the Decision Table there can be taken into account other factors such as importance of some attributes not expressed in their functional dependencies [21].

Once some relative reduct is chosen, the Decision Table contains only necessary attributes, yet still not all their values are necessarily needed for the classification process to be performed. Hence there follows another stage of reduction of the Decision Table and it is employing the concept of relative value reduct or D-value reduct and the core of relative value reducts or D-value core.

It is said that a value of attribute $c \in C$ is D-dispensable for $x \in U$ if

$$C(x) \subseteq D(x) \Rightarrow C_c(x) \subseteq D(x) \tag{6}$$

otherwise the value of attribute c is D-indispensable for x. If for every attribute $c \in C$ value of c is D-indispensable for x, then C is called D-independent for x.

Subset $C' \subseteq C$ is a relative value reduct (D-reduct) of C for x if and only if C' is D-independent for x and

$$C(x) \subseteq D(x) \Rightarrow C'(x) \subseteq D(x) \tag{7}$$

The set of all D-indispensable for x values of attributes in C is called the reletive value core (D-core) of C for x and denoted by $CORE_D^x(C)$, with the property

$$CORE_D^x(C) = \bigcap RED_D^x(C) \tag{8}$$

where $RED_D^x(C)$ is the family of all D-reducts of C for x.

While, as before for the core of relative reducts, also the core of relative value reducts is composed of these values for attributes that have always to be present to maintain classification properties, usually relative value reducts are even more numerous than relative reducts and as a result of this the considerations on their selection lead to various versions of the decision algorithm that is constructed from extracted decision rules. The decision algorithm is comprised of conditional clauses and not only their number is the direct result of the previously selected relative value reducts but even to some extent the accuracy of classification when the decision algorithm is applied to testing data.

In the study of relative value reducts numerous factors and approaches can be considered, such as, for example, algorithms dedicated to the problem of finding coverage, described in the next section.

IV. Optimality Considerations

Optimisation is a multidimensional problem, each dimension corresponding to one optimality criterion.

In the considered space a point is called a Pareto point if there is no other point which:

- could better satisfy at least one criterion,
- at least of the same merit at satisfying other optimality criteria.

A Pareto point corresponds to the global optimum in single dimensional space of optimisation [22]. With multiple optimality criteria there can be multiple Pareto points, none of them being the Pareto point in the global discrete space of optimisation. With multiple Pareto points it is worthwhile to consider each optimum found respectively for each of the previously defined criteria [11].

Nowadays optimisation is not only present in all areas of science but it is generally considered itself [23] to be a branch of science and it encompasses both exhaustive and heuristic algorithms that trade the global optimum for sub-optimum but obtained with lowered computational complexity and shortened execution time needed to find a solution.

The problem addressed in this paper is that of optimisation within decision rule extraction process necessary to construct the decision algorithm for classification with rough set-based approach. In such context the optimisation space can be considered as two-dimensional, with one optimality criterion being unsurprisingly the accuracy of classification and another the length of the decision algorithm expressed by the number of conditional clauses. Within this setup both optimality criteria are determined by the relative value reducts selected for decision rules defined by each row of the Decision Table and this choice can be helped with by application of algorithms dedicated to find coverage.

Algorithms that aim to find coverage, whether they be exhaustive or approximating [24], [25], are widely employed in many areas of science, for example in the tasks from graph theory (like graph colouring or finding maximal independent sets), or logic function minimisation (the choice of prime implicants). Finding the minimal cover can involve creating a logic expression describing the possible choices such as Petrick product, or building the coverage table or matrix whose columns and rows indicate how one selection is better than another by using the concept of dominance.

In the coverage table typically objects for which the cover is sought after are represented in rows while those covering are placed in columns and at the intersection the coverage is indicated by a check mark ($\surd$). The column dominates another column when it possesses not only those check marks as the

latter column but also some others. Dominated columns can be disregarded since they cannot offer better coverage than dominating ones. A column is essential when it contains at least one check mark which is the only one for some row. Essential columns create the core of all possible solutions - they must be present in any constructed cover. Thus essential columns also need no particular attention to be paid. It leaves only these columns that are neither essential nor dominated, and which give base to possibly several alternative solutions for minimal cover of the same merit [26].

If there exist several solutions of the same cardinality, there can be taken into account other (task-dependent) factors, in the considered context for example the number of attributes present in relative value reducts (the fewer the better for the decision algorithm) or the support of relative value reducts (for how many decision rules they are valid). All these described approaches were tested in the stylometric authorship attribution task, details of which are given next.

V. Experiments

Training texts used in experiments come from 4 novels by two famous Polish writers, Henryk Sienkiewicz ("Potop" and "Krzyżacy") and Boleslaw Prus ("Lalka" and "Faraon"), 4×9 samples from each novel. The testing rules (36) were based on the second set of 4 novels ("Rodzina Połanieckich" and "Quo vadis" by Sienkiewicz, and "Emancypantki" and "Placówka" by Prus). The samples were fragments of these novels of comparable length, chapters if possible.

The choice of novels as opposed to short works is motivated by the fact that wider corpora of texts provides more knowledge about individual writer's style and is more likely to result in higher classification ratio.

The Decision Table was created basing on occurrence frequencies of punctuation marks which belong to the group of syntactic descriptors that reflect the organisation of texts into sentences and their various types.

The considered 8 punctuation marks were: a comma, a semicolon, a full stop, a bracket, a quotation mark, an exclamation mark, a question mark, and a colon, and their frequencies constituted the set of conditional attributes.

Obviously frequencies are continuous values and not discrete and classical rough set approach deals only with discrete data sets, thus either some discretisation process was necessary to be applied to the input data or modified relations dedicated to continuous attributes [27]. The former approach chosen with the simplest imaginable discretisation that is thresholding returns binary data yet firstly the threshold value has to be selected. For this purpose there were used 2-quantiles for each of conditional attributes independently on others, as specified by the Table I.

As text samples were to be attributed to one out of two writers one decision attribute D was used, with $D = 1$ indicating Prus and $D = 0$ pointing to Sienkiewicz. Thus the Decision Table II for the D being set (the upper half of the table) describes works by Prus, while the reset state of D (the bottom half of the table) corresponds to works by Sienkiewicz. The total number of attributes was nine.

Table I
2-quantiles of frequencies

Attribute	Attribute median frequency
,	$MF_{\{,\}} = 0.101128$
;	$MF_{\{;\}} = 0.003055$
.	$MF_{\{.\}} = 0.110114$
(	$MF_{\{(\}} = 0.000128$
"	$MF_{\{"\}} = 0.003881$
!	$MF_{\{!\}} = 0.012082$
?	$MF_{\{?\}} = 0.010168$
:	$MF_{\{:\}} = 0.006575$

Table II
Decision Table

R	Conditional attributes								D
	,	;	.	(	"	!	?	:	
1	0	0	1	1	1	0	1	0	1
2	1	1	0	1	0	0	0	0	1
3	1	0	1	1	0	0	1	1	1
4	1	0	1	1	0	0	1	0	1
5	0	0	1	1	1	0	0	0	1
6	0	0	1	1	0	1	1	0	1
7	0	0	1	1	0	0	1	1	1
8	0	0	1	1	0	1	1	0	1
9	0	1	1	1	0	0	0	0	1
10	0	1	1	1	1	1	1	1	1
11	1	1	1	0	0	1	1	0	1
12	0	0	1	0	0	0	1	0	1
13	0	1	1	1	1	0	1	0	1
14	0	1	1	0	1	0	1	0	1
15	0	1	1	1	1	1	0	1	1
16	0	1	1	1	1	1	1	1	1
17	0	1	1	1	1	0	0	0	1
18	0	1	1	1	1	0	0	1	1
19	1	0	0	0	1	0	0	1	0
20	1	0	0	1	0	1	0	1	0
21	1	0	0	0	1	0	0	1	0
22	1	0	0	0	1	1	1	1	0
23	1	0	0	0	0	1	1	1	0
24	1	0	0	0	1	0	0	1	0
25	0	0	0	0	0	0	0	1	0
26	1	0	0	0	1	0	0	1	0
27	1	0	0	0	1	1	1	1	0
28	1	1	0	0	0	0	0	0	0
29	0	1	0	0	0	1	1	1	0
30	1	1	0	0	0	1	0	0	0
31	1	1	0	0	1	1	0	1	0
32	1	0	0	0	0	1	0	0	0
33	0	1	1	0	0	1	1	0	0
34	0	1	0	0	0	1	0	0	0
35	1	1	0	1	1	1	0	0	0
36	1	1	0	1	1	1	1	1	0

Once the Decision Table is constructed it needs to be tested for consistency that is whether there are no contradicting rules - with the same values of conditional attributes but different values of the decision attribute. Fortunately, there are no contradictions and the table is deterministic, thus its reduction could be attempted by finding relative reducts. Analysis returned several of them, with the core consisting of a comma and a bracket, as specified by the Table III.

In the choice of a relative reduct several approaches can be used [21], motivated for example by the significance of some of conditional attributes [9], yet in the considered application none of punctuation marks can be regarded as more important than others as there is no study of writer's style which would suggest that. Instead there is only considered the number of

Table III
GENERATED RELATIVE REDUCTS

	Conditional attributes				
RED_1	,	;	.	(	"
RED_2	,	;	(	"	?
RED_3	,	;	(	!	?
RED_4	,	.	(	!	
RED_5	,	(	!	?	:

conditional attributes within a reduct, that is the cardinality of obtained subsets of attributes.

The lowest cardinality has the 4th reduct on the list, the only one with four conditional attributes instead of 5 as it is in all other cases, thus the reduct containing a comma, a full stop, a bracket and an exclamation mark was selected for the following computations.

For the Decision Table limited to these attributes present in the chosen relative reduct, to all decision rules there was next applied the concept of relative value reducts, which returned the list of possible selections, as specified by the Table IV.

Table IV
GENERATED RELATIVE VALUE REDUCTS

Value reduct	Decision rule numbers
, .	3, 4, 11, 25, 29, 34
, (	1, 5, 6, 7, 8, 9, 10, 13, 15, 16, 17, 18
. (	1, 3, 4, 5, 6, 7, 8, 9, 10, 13, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34
. !	1, 3, 4, 5, 7, 9, 12, 13, 14, 17, 18, 20, 22, 23, 27, 29, 30, 31, 32, 34, 35, 36
(!	1, 2, 3, 4, 5, 7, 9, 13, 17, 18
, (!	19, 20, 21, 24, 26, 28, 29, 33, 34, 35, 36

While for some rules only single relative value reducts could be used (belonging to the value core) - for example for decision rule 2 only the relative value reduct consisting of a bracket and an exclamation mark is valid, for others there were several possibilities thus optimisation of this extraction process needed to be studied.

It is useful to notice that when reduced to the part corresponding to the selected relative reduct, some of resulting decision rules were repeated - that is appeared more than once in the Decision Table. Whether a decision rule occurs once or many times, each occurrence results in the same set of possible relative value reducts to be selected for it, thus such repetitions at this stage can be considered together and not one by one.

This reasoning leads to the Decision Table V with the reduced number of rows, where for further reference simplicity the rules are enumerated once again.

For twelve decision rules of Table V next there have to be studied available relative value reducts, as listed by the following Table VI. At the bottom of the table there are also listed supports for each relative value reduct.

The close look at the Table VI reveals that the column for VR_2 is dominated by VR_4 (it also has significantly lower support), and that VR_1, VR_3, VR_5 and VR_6 are essential and thus these relative value reducts correspond to the value core. Yet the value core is insufficient since it does not provide the cover of all decision rules within the table. Hence, if VR_2 is disregarded as dominated, it leaves VR_4 together with the value core to give coverage for all decision rules.

Table V
REDUCED DECISION TABLE

	Reduct				
Decision rule	,	.	(	!	D
dr_1: 1, 5, 7, 9, 13, 17, 18	0	1	1	0	1
dr_2: 2	1	0	1	0	1
dr_3: 3,4	1	1	1	0	1
dr_4: 6, 8, 10, 15, 16	0	1	1	1	1
dr_5: 11	1	1	0	1	1
dr_6: 12, 14	0	1	0	0	1
dr_7: 19, 21, 24, 26, 28	1	0	0	0	0
dr_8: 20, 35, 36	1	0	1	1	0
dr_9: 22, 23, 27, 30, 31, 32	1	0	0	1	0
dr_{10}: 25	0	0	0	0	0
dr_{11}: 29, 34	0	0	0	1	0
dr_{12}: 33	0	1	0	1	0

Table VI
DECISION RULES AND THEIR RELATIVE VALUE REDUCTS

	VR_1	VR_2	VR_3	VR_4	VR_5	VR_6
DR	, .	, (	, (!	. (	. !	(!
dr_1		√		√	√	√
dr_2						√
dr_3		√		√	√	√
dr_4		√		√		
dr_5	√					
dr_6					√	
dr_7			√	√		
dr_8			√		√	
dr_9				√	√	
dr_{10}	√			√		
dr_{11}	√		√	√	√	
dr_{12}			√			
Support	6	12	11	28	22	10

However, still for most of decision rules there is required the choice which particular relative value reduct from those available should be used. In these considerations there can be taken into account the support of each of relative value reducts. It can be argued that the higher the support the stronger the reduct and this line of reasoning leads to the list of selections as presented by the Table VII.

Table VII
SELECTED RELATIVE VALUE REDUCTS

Value reduct				Decision rule numbers
VR_1	,	.		5
VR_3	,	(	!	12
VR_4	.	(		1, 3, 4, 7, 9, 10, 11
VR_5	.	!		6, 8
VR_6	(	!		2

Next these relative value reducts are applied to decision rules and after removing repetitive rows from the tabular form of decision algorithm DA_1 obtained is presented by the Table VIII.

The algorithm to be constructed will comprise the total of seven conditional clauses, four for the decision attribute being set and three for the reset state of D and its classification accuracy needs to be tested next.

However, before testing yet another matter needs to be studied. In the methodology presented above the construction

Table VIII
DECISION ALGORITHM DA_1

Attributes				
,	.	(	!	D
	1	1		1
		1	0	1
1	1			1
	1		0	1
	0	0		0
	0		1	0
0		0	1	0

of the outcome decision algorithm is divided in two phases, the first of which is selecting relative value reducts for decision rules and the second being application of these selected reducts to rules. The problem is that the division of the whole process does not guarantee the minimal solution in terms of the number of conditional clauses. This is due to the fact that relative value reducts should not be discussed outside the context of decision rules for which they are valid.

For all considerations and computations it is most advantageous to remember what relative value reducts really are: they are masks put over decision rules and the same relative value reduct applied to two different decision rules can result in two different clauses for the decision algorithm. Therefore the selection of relative value reducts by themselves cannot constitute the immediate and direct answer to conditional clauses extracted for the decision algorithm.

Described line of argument results in conclusion that relative value reducts should be considered not only by themselves but also with taking into account what is the outcome of their application to all those decision rules for which they are valid, and such reasoning leads to obtaining the table with all conditional clauses that can be possibly created through all available choices of reducts. This for the considered example is presented in the Table IX.

Table IX
CONDITIONAL CLAUSES OBTAINED THROUGH VRS

	Attributes				
R	,	.	(	!	D
g		1		0	1
h	1	1			1
i			1	0	1
j		1	1		1
k	0		1		1
l	1		1		1
a	1		0	0	0
b		0	0		0
c	1		1	1	0
d		0		1	0
e	0	0			0
f	0		0	1	0

If all clauses were included, the decision algorithm DA_2 would consist of 6 rules for the decision attribute equal 1, and 6 for the decision attribute being 0, giving the total of 12 conditional clauses.

Such algorithm, although reflecting characteristics of the training set in the most detailed way does not promise to be more effective at generalising property than others, and some shorter can be searched for. In this search the clauses for different values of decision attribute are studied separately and Table X and XI specify out of which decision rules they come from.

Table X
CHOICE OF CONDITIONAL CLAUSES FOR $D = 1$

	Value reducts					
DR	g	h	i	j	k	l
dr_1	√		√	√	√	
dr_2			√			
dr_3	√		√	√		√
dr_4				√	√	
dr_5		√				
dr_6	√					

From all six columns g, h and i are essential and column l is dominated by i, which makes the choice necessary only between columns j and k. Yet j dominates k thus the former can be selected.

Table XI
CHOICE OF CONDITIONAL CLAUSES FOR $D = 0$

	Value reducts					
DR	a	b	c	d	e	f
dr_7	√	√				
dr_8			√	√		
dr_9		√		√		
dr_{10}		√			√	
dr_{11}		√		√	√	√
dr_{12}						√

In the Table XI only column f is essential, and b dominates a and e. As c is dominated by d, that leaves columns b, d and f to constitute the cover.

Another approach to finding a cover is offered by the previously mentioned Petrick product, which relies on construction of a logic expression that reflects possible choices of relative value reducts for decision rules. Since each decision rule must be covered the product consists of that many sums as many rules there are and each sum indicates possible reducts. There is no point in considering both parts of the decision table, that is for the decision attribute being set and reset, together. Instead for $D = 1$ there is created the following product

$$(g + i + j + k)i(g + i + j + l)(j + k)hg = 1$$

and for $D = 0$

$$(a + b)(c + d)(b + d)(b + e)(b + d + e + f)f = 1$$

These products are next transformed accordingly to the postulates and theorems of Boolean algebra with its inclusion rules. The first product returns

$$\begin{aligned} ghi(g + i + j + kl)(j + k) &= 1 \\ (ghij + ghik)(g + i + j + kl) &= 1 \\ ghij + ghijkl + ghik + ghijk + ghikl &= 1 \\ ghij + ghik &= 1 \end{aligned}$$

and the other

$$\begin{aligned} (a + b)(c + d)(b + de)f &= 1 \\ (ac + ad + bc + bd)(b + de)f &= 1 \\ (abc + abd + bc + bd + acde + ade + bcde + bde)f &= 1 \\ bcf + bdf + adef &= 1 \end{aligned}$$

That results in $2 \times 3 = 6$ different versions of decision algorithm, four with the number of conditional clauses equal 7 and two with eight clauses as follows:

$\{g, h, i, j \mid b, c, f\}$ - DA_3, Table XII

$\{g, h, i, j \mid b, d, f\}$ - DA_1,
$\{g, h, i, j \mid a, d, e, f\}$ - DA_4, Table XIII
$\{g, h, i, k \mid b, c, f\}$ - DA_5, Table XIV
$\{g, h, i, k \mid b, d, f\}$ - DA_6, Table XV
$\{g, h, i, k \mid a, d, e, f\}$ - DA_7, Table XVI.

The first version is indexed with 3 and the second with 1 for this reason that actually this second algorithm corresponds to the one previously obtained and presented in the Table VIII, and the second version is the one composed from all conditional clauses, given by the Table IX.

Table XII
DECISION ALGORITHM DA_3

	Attributes				
R	,	.	(	!	D
g		1		0	1
h	1	1			1
i			1	0	1
j		1	1		1
b		0	0		0
c	1		1	1	0
f	0		0	1	0

Table XIII
DECISION ALGORITHM DA_4

	Attributes				
R	,	.	(	!	D
g		1		0	1
h	1	1			1
i			1	0	1
j		1	1		1
a	1		0	0	0
d		0		1	0
e	0	0			0
f	0		0	1	0

Table XIV
DECISION ALGORITHM DA_5

	Attributes				
R	,	.	(	!	D
g		1		0	1
h	1	1			1
i			1	0	1
k	0		1		1
b		0	0		0
c	1		1	1	0
f	0		0	1	0

Table XV
DECISION ALGORITHM DA_6

	Attributes				
R	,	.	(	!	D
g		1		0	1
h	1	1			1
i			1	0	1
k	0		1		1
b		0	0		0
d		0		1	0
f	0		0	1	0

Next all these algorithms were subjected to testing, the results of which are presented and discussed in detail in the section that follows.

Table XVI
DECISION ALGORITHM DA_7

	Attributes				
R	,	.	(	!	D
g		1		0	1
h	1	1			1
i			1	0	1
k	0		1		1
a	1		0	0	0
d		0		1	0
e	0	0			0
f	0		0	1	0

VI. RESULTS AND DISCUSSION

Automatic knowledge processing technique applied to Tables VIII, IX, XII, XIII, XIV, XV, and XVI results in Decision Algorithms in which there were incorporated medians of frequencies previously used in the discretisation of the continuous input space. With such approach testing examples in fact do not have to be discrete.

All Decision Algorithms consist of two "If ...then ..." sentences, one per each value of the decision attribute D. The conditional sentences are composed of inequalities checking frequencies of attributes indicated by relative value reducts.

$DA_1 : \{g, h, i, j \mid b, d, f\}$
PRUS ($D = 1$) If:
$(F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{!\}} < MF_{\{!\}})$ OR
$(F_{\{,\}} \geq MF_{\{,\}}$ AND $F_{\{.\}} \geq MF_{\{.\}})$ OR
$(F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}})$ OR
$(F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}})$
SIENKIEWICZ ($D = 0$) If:
$(F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}})$ OR
$(F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{!\}} \geq MF_{\{!\}})$ OR
$(F_{\{,\}} < MF_{\{,\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}})$

DA_2 : complete set
PRUS ($D = 1$) If:
$(F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{!\}} < MF_{\{!\}})$ OR
$(F_{\{,\}} \geq MF_{\{,\}}$ AND $F_{\{.\}} \geq MF_{\{.\}})$ OR
$(F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}})$ OR
$(F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}})$ OR
$(F_{\{,\}} < MF_{\{,\}}$ AND $F_{\{(\}} \geq MF_{\{(\}})$ OR
$(F_{\{,\}} \geq MF_{\{,\}}$ AND $F_{\{(\}} \geq MF_{\{(\}})$
SIENKIEWICZ ($D = 0$) If:
$(F_{\{,\}} \geq MF_{\{,\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}})$ OR
$(F_{\{,\}} < MF_{\{,\}}$ AND $F_{\{(\}} < MF_{\{(\}})$ OR
$(F_{\{,\}} \geq MF_{\{,\}}$ AND $F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}})$ OR
$(F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{!\}} \geq MF_{\{!\}})$ OR
$(F_{\{,\}} < MF_{\{,\}}$ AND $F_{\{.\}} < MF_{\{.\}}$ OR
$(F_{\{,\}} < MF_{\{,\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}})$

$DA_3 : \{g, h, i, j \mid b, c, f\}$
PRUS ($D = 1$) If:
$(F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{!\}} < MF_{\{!\}})$ OR
$(F_{\{,\}} \geq MF_{\{,\}}$ AND $F_{\{.\}} \geq MF_{\{.\}})$ OR
$(F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}})$ OR
$(F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}})$
SIENKIEWICZ ($D = 0$) If:
$(F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}})$ OR
$(F_{\{,\}} \geq MF_{\{,\}}$ AND $F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}})$ OR
$(F_{\{,\}} < MF_{\{,\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}})$

$DA_4 : \{g, h, i, j \mid a, d, e, f\}$
PRUS ($D = 1$) If:
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{.\}} \geq MF_{\{.\}}$) OR
($F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}}$)
SIENKIEWICZ ($D = 0$) If:
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{.\}} < MF_{\{.\}}$ OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$)
$DA_5 : \{g, h, i, k \mid b, c, f\}$
PRUS ($D = 1$) If:
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{.\}} \geq MF_{\{.\}}$) OR
($F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}}$)
SIENKIEWICZ ($D = 0$) If:
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$) OR
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$)
$DA_6 : \{g, h, i, k \mid b, d, f\}$
PRUS ($D = 1$) If:
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{.\}} \geq MF_{\{.\}}$) OR
($F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}}$) OR
SIENKIEWICZ ($D = 0$) If:
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$)
$DA_7 : \{g, h, i, k \mid a, d, e, f\}$
PRUS ($D = 1$) If:
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{.\}} \geq MF_{\{.\}}$) OR
($F_{\{(\}} \geq MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} \geq MF_{\{(\}}$)
SIENKIEWICZ ($D = 0$) If:
($F_{\{.\}} \geq MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} < MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$) OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{.\}} < MF_{\{.\}}$ OR
($F_{\{.\}} < MF_{\{.\}}$ AND $F_{\{(\}} < MF_{\{(\}}$ AND $F_{\{!\}} \geq MF_{\{!\}}$)

The results obtained in the research performed are given in the Table XVII into categories of classification verdicts for all testing samples treated individually: as correct, incorrect and undecided, along with accuracy of classification and a number of conditional clauses in the algorithm.

Table XVII
CLASSIFICATION RESULTS

Classification verdict	Decision Algorithms				
	DA_1	DA_2	$DA_{3,5,6}$	DA_4	DA_7
correct	31	31	31	30	31
incorrect	5	5	4	5	5
undecided	0	0	1	1	0
accuracy (%)	86.11	86.11	86.11	83.33	86.11
no of rules	7	12	7	8	8

In all cases the classification accuracy can be considered satisfactory, yet obviously the higher one of 86.11% is preferable (when compared to accuracy of 83.33% it reduces the classification error by 16.67%). The lowest number of correct classification verdicts for testing samples belongs with DA_4 which also has the lowest accuracy. On the other hand the lowest number of incorrectly classified samples happens for DA_3, DA_5, and DA_6.

The length of the decision algorithm constructed varies significantly from the maximum of 12 clauses when all conditional clauses are included in it to the minimum of 7, which is reduction by 41.66%.

The overall classification accuracy can be presented as classification of whole novels to be attributed as specified by the Table XVIII.

Table XVIII
CLASSIFICATION RESULTS FOR THE WHOLE NOVELS

Author	Text	$DA_{1,2,3,5,6}$	DA_4	DA_7
Prus	"Emancypantki"	100%	100%	100%
	"Placówka"	88.9%	77.8%	100%
Sienkiewicz	"Rodzina Połanieckich"	88.9%	88.9%	88.9%
	"Quo vadis"	66.7%	66.7%	66.7%
	Avarage	86.1%	83.3%	88.9%

It is worth noticing that for the whole novels classification results vary for different versions of decision algorithms just for one out of four tested novels, namely for "Placówka", while for the other three total results are the same.

In cases of some group of testing samples with significantly lower classification accuracy than others it may be advantageous to reconsider whether these troublesome samples should be included in the testing set in the first place. Evidently one novel has some characteristic not shared by others thus it would be better to include it within the training set instead. As it is the differences are significant enough to range the average classification accuracy from 83.3% to 88.9%.

When obtained results are plotted in the 2-dimensional optimisation space being considered the case of classification for individual samples is depicted in Fig. 1. In this chart there exists the Pareto point in the space and it is for decision algorithms DA_1, DA_3, DA_5 and DA_6 since all consist of seven conditional clauses and give classification ratio of 86.11%.

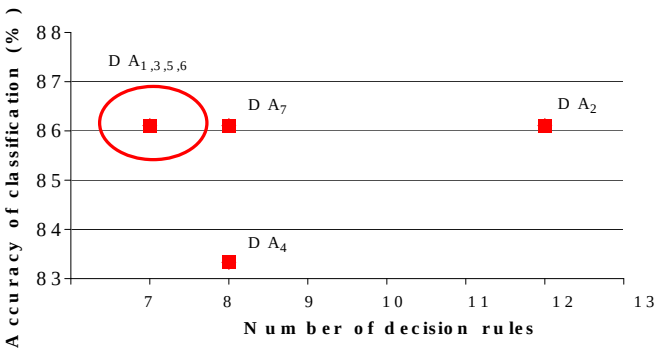

Figure 1. Optimisation space with classification accuracy considered for individual testing samples

On the other hand, when average ratio of classification is studied, as shown in Fig. 2, it becomes clear that there is no global Pareto point in the optimisation space.

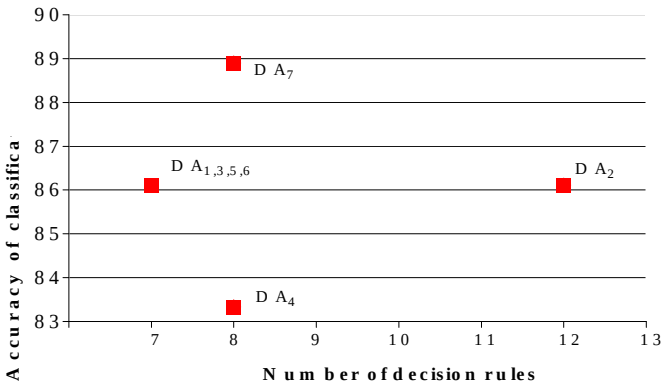

Figure 2. Optimisation space with classification accuracy considered for whole novels

While the highest average classification accuracy of 88.9% is offered by the decision algorithm DA_7, the minimal number of conditional clauses included, which is seven, belongs with decision algorithms DA_1, DA_3, DA_5 and DA_6, and DA_7 consists of eight clauses.

VII. Conclusion

Presented results of optimisation for decision rule extraction within the rough set-based authorship attribution of literary texts were satisfactory since in case of studying classification results by individual testing samples for both considered optimality criteria, that is the classification accuracy and the number of conditional clauses included in a decision algorithm, some improvement was achieved.

Yet the applied methodology was manageable only due to the high degree of reduction of the constructed Decision Table. If there were many more decision rules the whole process would get cumbersome and other systematic coverage procedures should be used, for example from Espresso system.

Furthermore in future research the number of dimensions for the optimisation space can be increased for example by incorporating such criteria as quantities of conditional attributes within all clauses which comprise a decision algorithm. Since each attribute means comparison of values while testing, fewer number of clauses with higher number of comparisons does not necessarily mean faster processing than with more clauses that require fewer comparisons.

Acknowledgment

The software used to calculate frequencies of punctuation marks for texts was implemented by Mr P. Cichoń under supervision of Professor K.A. Cyran, in fulfillment of requirements for M.Sc. thesis submitted at the Faculty of Computer Science, the Silesian University of Technology, Gliwice, Poland.

References

[1] G. Bonanno and F. Moschella and S. Rinaudo and P. Pantano and V. Talarico, Manual and evolutionary equalization in text mining, *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, 2007, pp. 262–267.

[2] S. Argamon and J. Karlgren and J.G. Shanahan, eds., Stylistic analysis of text for information access, *Proceedings of the 28th International ACM Conference on Research and Development in Information Retrieval*, Brazil, 2005.

[3] A. Lampropoulos and E. Galiotou and I. Manolessou and A. Ralli, A finite-state approach to the computational morphology of early modern Greek, *Proceedings of the 7th WSEAS International Conference on Applied Computer Science*, 2007, pp. 242–245.

[4] A. Caballero and K. Yen and Y. Fang, Classification with diffuse or incomplete information, *WSEAS Transactions on Systems and Control* 3(6), 2008, pp. 617–626.

[5] L.A. Zadeh, A fuzzy-algorithmic approach to the definition of complex or imprecise concepts, *International Journal on Man-Machine Studies* 8, 1976, pp. 249-291.

[6] Z. Pawlak, Rough Set Rudiments, *Institute of Computer Science Report, Warsaw University of Technology, Poland*, 1996, pp. 1–47.

[7] Q. Shen, Rough feature selection for intelligent classifiers, *LNCS Transactions on Rough sets* 7, 2006, pp. 244–255.

[8] M. Doumpos and A. Salappa, Feature selection algorithms in classification problems: an experimental evaluation, *WSEAS Transactions on Information Science & Applications* 2(2), 2005, pp. 77–82.

[9] M.J. Moshkov and M. Piliszczuk and B. Zielosko, On Partial Covers, Reducts and Decision Rules with Weights, *Transactions on Rough Sets* 6, 2006, pp. 211–246.

[10] J. Stefanowski, On Combined Classifiers, Rule Induction and Rough Sets, *Transactions on Rough Sets* 6, 2006, pp. 329–350.

[11] G. De Micheli, Synthesis and optimization of digital circuits, *McGraw-Hill*, New York, USA, 1994.

[12] R.D. Peng and H. Hengartner, Quantitative analysis of literary styles, *The American Statistician* 56(3), 2002, pp. 15–38.

[13] D.T. Tran and T.D. Pham, Markov and fuzzy models for written language verification, *WSEAS Transactions on Systems* 4(4), 2005, pp. 268–272.

[14] D.V. Khmelev and F.J. Tweedie, Using Markov chains for identification of writers, *Literary and Linguistic Computing* 16(4), 2001, pp. 299–307.

[15] W. Buckland, Forensic semiotics, *The Semiotic Review of Books* 10(3), 1999.

[16] R.A.J. Matthews and T.V.N. Merriam, Distinguishing literary styles using neural networks, in E. Fiesler and R. Beale, eds., *Handbook of neural computation,* Oxford University Press, 1997, pp. G8.1.1–6.

[17] J.F. Jimenez and F.J. Cuevas and J.M. Carpio, Genetic algorithms applied to clustering problem and data mining, *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, 2007, pp. 219–224.

[18] P. Jirava and J. Krupka, Classification model based on rough and fuzzy sets theory, *Proceedings of the 6th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, 2007, pp. 198–202.

[19] J. Krupka and P. Jirava, Modelling of rough-fuzzy classifier, *WSEAS Transactions on Systems* 7(3), 2008, pp. 140–149.

[20] R.B. Perez and A. Nowe and P. Vrancx and Y. Gomez and D.Y. Caballero, Using Ant Colony Optimization and rough set theory to feature selection, *WSEAS Transactions on Information Science & Applications* 2(5), 2005, pp. 512–517.

[21] M.J. Moshkow and A. Skowron and Z. Suraj, On Covering Attribute Sets by Reducts, in M. Kryszkiewicz and J.F. Peters and H. Rybinski and A. Skowron, eds., *Lecture Notes in Artificial Intelligence* 4585, 2007, pp. 175–180.

[22] T.-Y. Chen and Y.L. Cheng, Global Optimization using Hybrid Approach, *WSEAS Transactions on Mathematics* 7(5), 2008, pp. 60–69.

[23] A. Swierniak and A. Galuszka, Optimization methods and decision making - lecture notes, *Publishers of the Silesian University of Technology*, Gliwice, Poland, 2003.

[24] P. Slavik, Approximation algorithms for set cover and related problems, *Ph.D. Thesis, University of New York*, Buffalo, 1998.

[25] P. Slavik, A tight analysis of the greedy algorithm for set cover, *Proceedings of 28th Annual ACM Symposium on the Theory of Computing*, 1996, pp. 435–441.

[26] U. Stańczyk and K.A. Cyran and B. Pochopień, Theory of logic circuits Vol. 1 Fundamental issues, *Publishers of the Silesian University of Technology*, Gliwice, Poland, 2007.

[27] K.A. Cyran and U. Stanczyk, Indiscernibility relation for continuous attributes: application in image recognition, *Lecture Notes in Artificial Intelligence* 4585, 2007, pp. 726 - 735.

Urszula Stanczyk received her M.Sc. degree in computer science from the Silesian University of Technology, Gliwice, Poland in 1993. In 2003 she received her Ph.D. degree (with honours) in technical sciences with specialty in computer science from the same University. Her Ph.D. dissertation addresses the issues of applying some elements of logic circuits theory and techniques to optimisation of pre-processing of binary images.

From 1993 till 2000 she was a teaching assistant, from 2000 till 2003 a lecturer, and from 2004 till present an assistant professor in the Division of Microinformatics and Automata Theory at the Institute of Informatics, SUT. In 2004 and 2005 she was a lecturer in the Gliwice branch of the Academia of Polonia in Czestochowa, Poland. From 2004 Dr Stanczyk has been the Editor-in-Chief of the Activity Report for the Institute of Informatics.

Her scientific research interests include digital image processing and recognition, with special emphasis on mathematical morphology methods, computational intelligence and especially rough set theory and artificial neural networks, stylometry and its tasks, elements of theory of logic circuits, their design procedures and optimisation of implementations, as well as arithmetic of digital systems.