Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 3273–3282

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Assessing quality of decision reducts

Urszula Stańczyk[a,*], Beata Zielosko[b]

[a]*Department of Graphics, Computer Vision and Digital Systems, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 2A, 44-100 Gliwice, Poland*

[b]*Institute of Computer Science, University of Silesia in Katowice, Będzińska 39, 41-200 Sosnowiec, Poland*

Abstract

The paper presents research focused on decision reducts, a feature reduction mechanism inherent to rough sets theory. As a reduct enables to protect the discriminative properties of attributes with respect to described concepts, from the point of data representation, a reduct length is considered to be the most important measure of its quality. However, such approach is insufficient while taking into account the performance of a reduct-based rule classifier applied to test samples. When many reducts of the same length are available, they can lead to vastly different predictions. The paper provides a description for the proposed procedure for iterative reduct generation, which results in decrease of diversity in the observed levels of accuracy, supporting reduct selection. The procedure was applied for binary classification with balanced classes, for the stylometric task of authorship attribution.

Keywords: reduct; rough sets; decision rules; classification; stylometry; feature reduction

1. Introduction

Reduct is one of the fundamental concepts of rough set theory (RST), a data mining approach that supports operating on inconsistent and uncertain information, invented by Z. Pawlak [13]. For a special case of information systems, called a decision table, a reduct is such minimal subset of attributes that allows to construct a classification model with the same predictive power as for the entire set of available features. The cardinality of a reduct defines its length. Depending on data, many reducts can be found for a single decision table. Some of attributes occur in the constructed reducts more often than others, some appear in reducts of specific cardinalities. A set of reducts can be considered as an additional source of knowledge on attributes, and employed to support a task of feature characterisation [18].

The area of reduct generation has been widely studied, which resulted in a variety of algorithms employed in search for single reducts or their groups, or reducts with specific parameters [11]. Reducts can be obtained by an exhaustive algorithm, yet such process is computationally demanding. The fact that it guarantees locating all available reducts

* Corresponding author.
E-mail address: urszula.stanczyk@polsl.pl

1877-0509
10.1016/j.procs.2020.09.121

can be seen as both an advantage and disadvantage. With many reducts to choose from, some additional criteria for reduct selection are required. Genetic algorithms [20] work significantly faster. They return a subset of reducts, which could be even smaller than the requested number depending on suitability of tested candidates. Greedy heuristics often concentrate the search on a certain factor, such as providing coverage, or optimisation with respect to length [1, 21].

From the point of view of data representation the quality of reducts is usually measured in terms of cardinality. Smaller reducts are preferred, as they offer a mechanism for dimensionality reduction. Fewer features lead to decreased storage requirements, simplification of an analysis and understanding, and can result in enhanced interpretability. However, when a reduct is used as a base for construction of a classification model, limiting considerations to reduct lengths is insufficient for an informed reduct selection. The difference in reduct size by a single variable can cause a great difference in accuracy, even two reducts of the same cardinality can significantly vary in their predictive powers.

In the paper a novel approach to reduct quality assessment is presented, dedicated to the reduction of diversity of classification outcomes for reduct-based rule classifiers. The procedure for generation of reducts is executed repetitively for gradually decreasing subsets of attributes, and to each reduct a survival index is assigned, indicating the number of iterations in which it is found. The features selected to be discarded are chosen by the rate of occurrence in the constructed reducts. The process stops when all attributes are included in reducts in the same degree.

The proposed approach was validated experimentally. Firstly, a group of reducts was found by a genetic algorithm. Secondly, an exhaustive search for reducts was executed. To the sets including all reducts the iterative procedure was employed, focused on reduction of attributes and limitation of the number of reducts. Thirdly, the set of reducts from the genetic algorithm was extended, by adding selected elements found in exhaustive search, to form a representative sample with respect to reduct length and assigned survival index. For all these reducts new decision tables were obtained and decision rules induced with classical rough set approach [12]. The performance of all classification systems was evaluated with tests sets. The results from the experiments showed that with increasing numbers of iterations it was possible to construct such reducts that led to rule classifiers with much closer predictive powers.

The datasets used in research works were prepared for a stylometric task of binary authorship attribution. Authors were compared by their writing styles, and these styles were defined by selected linguistic descriptors [15], which reflect individual preferences and traits, habits of sentence formulation. Depending on type, they can be grouped into many categories. With samples of known authorship available for comparison, confirmation or rejection of authorship for disputed cases can be treated as a classification task, to be solved with supervised learning approaches [9].

The paper includes Section 2 with description of fundamental notions of rough set theory and popular approaches to reduct construction. Section 3 presents the proposed iterative procedure for reduct generation. The framework of experiments and obtained results are commented in Section 4. Section 5 concludes the paper.

2. Fundamental concepts of rough sets theory

In the rough set theory [13], the form that is used for data representation is called a *data table*, which is a special case of an information system. Its columns are labelled by the attributes, rows by objects, and entries of the table provide attribute values. The attributes are divided into two groups, named condition and decision respectively. In many studied problems single decision attributes exist. Then the table is defined as $DT = (U, A \bigcup \{d\})$. U is the universe—a non-empty, finite set of objects, and $A = \{a_1, \ldots, a_m\}$ is a non-empty, finite set of condition attributes. The set of values for the decision attribute d is given by $V_d = \{d_1, \ldots, d_{|V_d|}\}$, and for an attribute a_i its set of values is V_a, $a_i : U \rightarrow V_a$. Based on data included in a decision table, *decision rules* can be constructed. For $1 \leq i_1 < \ldots < i_k \leq m, v_i \in V_{a_i}$, and $1 \leq v_d \leq |V_d|$, the rules are presented in the form $(a_{i_1} = v_1) \wedge \ldots \wedge (a_{i_k} = v_k) \rightarrow d = v_d$.

One of the important notions of the rough set theory is *indiscernibility relation*, generated by information about objects of the universe. If such objects are characterised by the same values of attributes, they are indiscernible from the point of view of the available knowledge about them, and they are constitute granules of knowledge about the universe. In rough sets, imprecise concepts are approximated based on clusters of indiscernible objects, i.e, any imprecise concept is replaced by a pair of precise ones called the lower and the upper approximation of the rough concept. Boundary region (which is a difference between the lower and upper approximation) expresses imprecision, i.e., if it is non-empty, it means that our knowledge about the concept is not sufficient to define it precisely.

Reducts belong to rudiments of rough sets. In the reported research, *decision reducts* were investigated. $B \subseteq A$ is a decision reduct for DT if and only if it is an irreducible subset of condition attributes such that each pair of objects

$x, y \in U$ satisfying inequality $d(x) \neq d(y)$ is discerned by B. There are different types of decision reducts, among others: dynamic [2], maximally discerning [10], based on the generalised decision [12], decision bireducts [19].

In the case of decision reducts, knowledge discovery is the most popular application domain, where attributes included in a reduct form a base for construction of a model, which is evaluated by the accuracy of classification. For knowledge representation a reduct length is all-important, because it projects into interpretability and understanding of a model based on the reduct. However, if a decision algorithm is obtained based on reducts that are merely short, it does not always lead to a higher classification accuracy than in the case of using other reducts. To improve this situation, a dynamic approach to reduct construction was proposed [4]. It uses stability as a criterion denoting frequency with which reducts of a decision table occur as reducts also in subsets of that decision table. If the frequency corresponds to or exceeds a certain percentage of the subsets, such reducts are called stable.

Reducts can be constructed in many ways. In the case of decision bireducts, a subset of attributes can be evaluated by means of a subset of objects, for which it assures some satisfactory classification accuracy. Information about objects, whose decisions are properly described, allows to verify whether classifiers, designed using different subsets of attributes, do not repeat mistakes on the same areas of the training data. An another popular algorithm, which allows to find all reducts, is based on a discernibility matrix [14]. The rows and columns of the matrix correspond to objects, and the element of the matrix is the set of all attributes that distinguish the corresponding pair of objects. The set of reducts corresponds to the set of prime implicants of the reduced disjunctive form of the discernibility function.

It is possible to show that, for any m, there is an information system with m attributes, with an exponential (w.r.t. m) number of reducts. Taking this into account, and the fact that the problem of finding a minimal reduct is NP-hard, various algorithms based on heuristics were investigated, which allow to compute many reducts in a reasonable time, for example based on genetic algorithms [20], greedy algorithms, ant colony optimisation [8], particle swarm optimisation [5], and many others.

3. Iterative procedure for generating reducts

The relation between a reduct or a group of reducts and attributes can be used both ways. An attribute can be characterised by reducts in which it does or does not occur [16], their numbers and cardinalities, but also a reduct is described by included attributes. Some of data dependencies captured in reducts are fragile and subject to change upon any modification (such as feature reduction) of the environment—decision table, while some are strong enough to still be present even after several changes made. The proposed procedure of iterative reduct generation relies on removing such weak links from considerations, by gradual rejection of features used least often in reduct construction. The block diagram illustrating the steps involved in the processing is shown in Fig. 1.

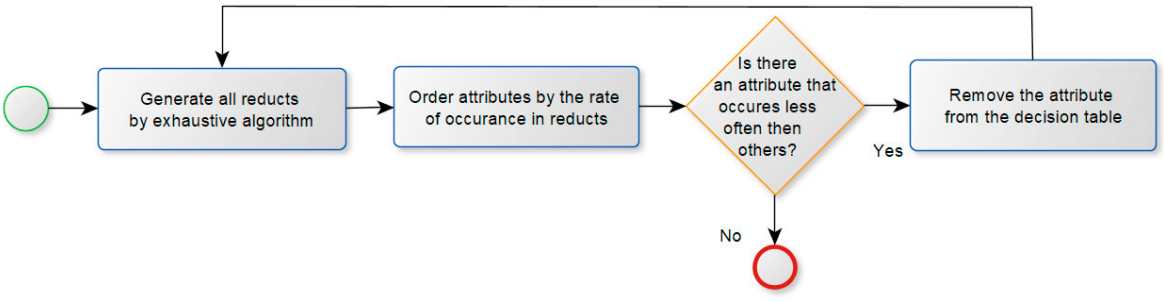

Figure 1. Block diagram for the iterative procedure of reduct generation.

The procedure starts with calculation of all reducts by exhaustive algorithm for the entire decision table. Next, the attributes are ordered by the number of reducts in which they are included. From the list of features the one that occurs least often in reducts is rejected, and a reduced decision table is constructed. With discarding of the attribute data dependencies change, and all previously found reducts containing this variable cease to be reducts. For the modified table once again all reducts are discovered, and the process of ordering and rejecting features is repeated. It stops when, for all attributes still taken into account, the numbers of reducts where they appear is the same.

Through this iterative procedure to each reduct a number is assigned, named reduct survival index (RedSI). For a reduct, RedSI states how many changes of the original decision table the reduct withstood intact. Since in each iteration a single attribute is removed from the table, a reduct survival index in this case corresponds to the number of features that can be discarded safely, without compromising the reduct ability of protecting predictive power.

The aim of the presented approach is to limit the number of reducts to such group of elements, which will be closer to each other in performance when used as a base for construction of a classification model. With increasing values of RedSI the numbers of calculated reducts decrease, and more and more of them relay on the attributes most helpful in discrimination of classes, which causes diminished dispersion of classification accuracy. The proposed procedure was validated through executed experiments, which are described in the next section.

4. Experimental validation

The experiments performed started with preparation of the input datasets, for which next groups of reducts where calculated by some algorithms, in particular by the proposed iterative generation procedure. Selected reducts were used for classification model construction, and their performance was evaluated with test sets. Obtained results were analysed with respect to reduct length and the discovered reduct survival index, as presented in detail below.

4.1. Characteristics of input datasets

A stylometric task of authorship attribution requires a definition of a writing style for an author, expressed by a set of linguistic descriptors. In the research works two pairs of authors were selected, two female writers (E. Wharton and M. Johnston) and two male writers (J. London and J. Curwood). Their works were divided into groups to provide base for the training and test samples. The novels were partitioned into parts of comparable size, over which occurrence frequency was calculated for a hundred selected function words and punctuation marks. Next, for feature reduction, several rankers implemented in WEKA [7] environment were employed, and 24 most relevant attributes selected:

after	any	before	by	during	never	such	that	there	whether	what	;
almost	around	but	how	on	same	until	then	though	within	who	,

After processing the training sets with 200 samples were obtained, and two test sets with 90 samples each, for two datasets: female and male writers. Real-valued features were transformed into categorical by discretisation with Fayyad and Irani supervised approach [6]. For female writers 4 out of 24 attributes were found as bringing no informative content with respect to classes, which resulted in assigning to them single bins for the whole range of values, and the same happened to 2 features for male writers. Thus the datasets were ready for rough set processing.

4.2. Reducts generated by genetic and exhaustive algorithms

For the prepared training data for both female and male writer dataset decision reducts were generated by two approaches: genetic algorithm [21] and exhaustive search, both implemented in Rough Sets Exploration System (RSES) [3] that was used in research. For the genetic algorithm the number of requested reducts was set to 200 for both datasets. The numbers of reducts with specific length found for all sets are listed in Table 1.

For the genetic algorithm cardinalities of reducts ranged from 4 for female, and 5 for male writers, to 10. There were more short reducts in the former case, and the same can be observed for reducts calculated by the exhaustive algorithm. The total number of reducts calculated by the exhaustive algorithm greatly varied between the two datasets. For both it was so high as to make the process of using all reducts to build classification models and testing them unfeasible.

For illustration of the processing executed by the proposed iterative procedure for reduct generation, a representative sample of reducts was chosen. To all reducts returned by the genetic algorithm, others were added from exhaustive search, to provide sufficiently high variety of lengths and discovered survival indices. All these selected reducts (300 for female writer dataset, and 500 for male writer dataset) served next as a base for induction of decision rules.

4.3. Reduct-based rule classifiers

All selected reducts were employed for feature reduction, and the original decision tables were modified correspondingly. For the limited decision tables, decision rules were inferred by exhaustive algorithm available in RSES

Table 1. Numbers of reducts of specific size generated by genetic and exhaustive algorithms for both datasets

Reduct length	4	5	6	7	8	9	10	11	12	13	Total
Genetic algorithm											
Female authors	1	18	60	57	37	24	3				200
Male authors		1	28	69	61	38	3				200
Exhaustive algorithm											
Female authors	1	22	117	228	236	191	65	26	28		914
Male authors		1	40	429	1257	2389	4196	3267	558	26	12163
Representative sample (reducts from genetic algorithm + selected cases from exhaustive algorithm)											
Female authors	1	20	74	84	61	41	10	4	5		300
Male authors		1	30	75	100	178	74	20	18	4	500

system. The rule sets were used in classification of test samples, with two strategies in case of occurring conflicts: simple voting, were each rule has a single vote, and standard voting, with weighting votes by rule support [17]. The averaged performance of rule classifiers based on reducts found by genetic algorithm is shown in Fig. 2 in the perspective of reduct cardinality as it is so often considered as the most important factor for reduct selection.

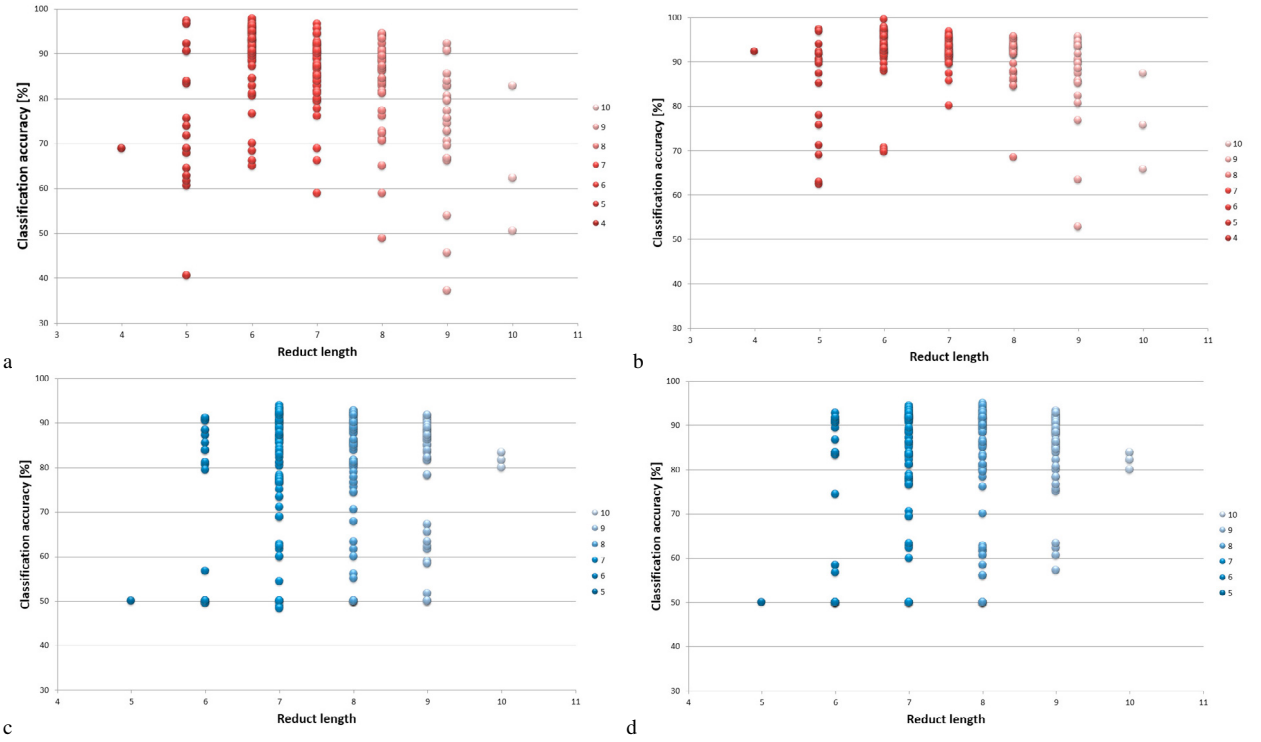

Figure 2. Performance of reduct-based rule classifiers [%] for reducts found by genetic algorithm, for: (a) and (b) female authors, (c) and (d) male authors, for (a) and (c) simple voting, and (b) and (d) standard voting strategy in case of conflicts.

It can be observed that female writer dataset led to higher levels of correct recognition of classes, whereas male authors proved to be more difficult. Weighted voting to some extent gave better results than simple voting strategy. It is clearly visible in all charts that minimal cardinality of reducts does not guarantee the best predictions. What is more, reducts with the same lengths led to some very wide ranges of classification accuracy, which was further evidenced by the averages and standard deviation calculated with respect to samples, as listed in Table 2.

Table 2. Averaged performance [%] and standard deviation for rule classifiers based on reducts generated by genetic algorithm

Reduct	Female authors		Male authors	
length	Simple voting	Standard voting	Simple voting	Standard voting
4	68.89 ± 20.43	92.22 ± 00.00		
5	75.06 ± 16.94	84.14 ± 12.13	50.00 ± 00.00	50.00 ± 00.00
6	90.05 ± 08.04	93.03 ± 06.49	68.53 ± 19.15	71.94 ± 19.29
7	86.29 ± 08.07	92.19 ± 03.27	78.60 ± 15.48	81.68 ± 14.45
8	83.27 ± 11.47	91.17 ± 05.38	78.17 ± 15.28	79.62 ± 16.07
9	73.89 ± 14.07	86.32 ± 10.32	78.07 ± 15.57	83.68 ± 12.01
10	65.19 ± 19.60	76.11 ± 11.57	81.67 ± 02.43	82.04 ± 03.15
Total	83.68 ± 13.09	90.36±8.46	76.85 ± 16.28	79.91 ± 15.72

When the same kind of analysis was applied to the extended group of reducts, formed by adding selected cases from exhaustive search to these returned by the genetic algorithm, most categories of reducts—corresponding to their varied lengths, included more representatives, yet the overall trends, both in the obtained averages and standard deviation, were the same, as shown in Table 3.

Table 3. Averaged performance [%] and standard deviation for rule classifiers based on a representative sample of reducts (generated by genetic algorithm + selected cases from exhaustive algorithm)

Reduct	Female authors		Male authors	
length	Simple voting	Standard voting	Simple voting	Standard voting
4	68.89 ± 20.43	92.22 ± 00.00		
5	74.92 ± 16.60	84.50 ± 11.57	50.00 ± 00.00	50.00 ± 00.00
6	89.91 ± 08.86	93.27 ± 06.02	69.72 ± 19.09	73.07 ± 19.14
7	86.70 ± 07.59	92.51 ± 03.43	77.87 ± 15.64	81.41 ± 14.76
8	84.16 ± 11.64	92.03 ± 04.64	79.46 ± 13.90	80.63 ± 14.39
9	76.60 ± 13.12	87.98 ± 08.48	79.72 ± 13.09	84.57 ± 10.43
10	76.28 ± 16.12	84.00 ± 09.61	78.36 ± 12.87	82.24 ± 11.50
11	72.64 ± 12.74	82.22 ± 10.66	73.64 ± 14.37	78.36 ± 13.15
12	79.17 ± 07.58	87.50 ± 02.38	74.26 ± 16.54	73.31 ± 16.52
13			84.31 ± 07.11	74.03 ± 12.52
Total	84.10 ± 12.05	90.95 ± 07.07	78.13 ± 14.64	81.46 ± 13.73

These observations confirmed the fact that regardless of the cardinality of a group of reducts taken into consideration, even when they are of the same length, any two examples cannot be reasonably expected to lead to some comparable performance of classification models constructed from them. This conclusion gave motivation for further research on a procedure for such categorisation of reducts that would result in some noticeable reduction of the dispersion of obtained predictive powers of reduct-based classifiers.

4.4. Iterative generation of reducts

The idea of the iterative procedure for reduct generation stems from perceiving calculated reducts as some form and representation of knowledge discovered in a decision table with respect to included attributes. Features that occur in many reducts can be seen as a part of some dominating patterns in data, and these patterns can be strong enough to still exist even when the table is reduced by removing some attributes. This line of reasoning leads to ordering of features, reflecting how often they are included in generated reducts, as a deciding factor for discarding attributes.

The starting point for the processing was generating reducts by the exhaustive algorithm. For both datasets the number of reducts found was so high, as making an analysis of all of them one by one impractical, and the choice—if guided solely by reduct lengths—risky. All attributes were next ordered based on the frequency of occurrence in reducts. For the first run of the procedure the differences in occurrences were rather high, in particular for male

writer dataset, where the total number of reducts was much higher than for female authors. As the result, the stopping criterion for the procedure—the absence of an attribute occurring less frequently than other variables—was obviously not met. These least frequent elements were removed from the decision tables, and the procedure of reduct generation repeated. The process continued as long as there were present some attributes to discard, which took 15 runs for female writer dataset and 14 for male authors. The observed ordering of features for all runs is listed in Tables 4 and 5.

Table 4. Ordering for attributes based on occurrence in reducts in each iteration for female writer dataset

Iteration																
0	1	2	3	4	5	6	7	8	9	10	11	12	13			14
,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	by
after	after	after	after	after	on	on	;	;	on	;	;	until	until	by	same	until
but	but	but	but	but	after	but	after	same	any	on	until	on	same	until	any	,
there	there	there	there	on	but	after	on	then	;	any	on	same	on	same	until	on
same	same	on	;	there	;	until	but	but	by	until	any	;	;	on	on	same
any	any	same	same	same	until	;	then	after	then	by	same	by	by	;	;	;
on	on	;	on	;	there	there	any	until	same	same	by	any	after	after	after	any
such	then	what	what	until	same	then	such	on	until	then	after	after				
;	;	any	then	then	such	never	until	by	after	after	then					
then	such	then	by	such	then	by	there	any	there	there						
what	what	such	until	by	by	same	same	there	but							
by	by	by	any	any	never	any	by	such								
before	before	before	who	what	what	such	never									
never	until	until	such	before	any	what										
until	never	never	before	never	before											
who	who	who	never	who												
how	around	around	around													
around	how	how														
whether	whether															
that																

The iteration denoted in tables as 0 corresponds to the initial generation of reducts for the entire sets of features, before any of them were rejected. The tables do not include the final run of the procedure, as they could not provide any particular ordering of variables, because they occurred exactly the same number of times in calculated reducts. In both tables for the 13th iteration three different orderings for attributes are listed. Such branching off was a consequence of some possible choices made at 12th iteration: for both datasets there were three features appearing in reducts less frequently than others, with the same numbers of occurrence. Instead of some arbitrary choice of one element out of these three to reject, all possible paths were tested.

Table 5. Ordering for attributes based on occurrence in reducts in each iteration for male writer dataset

Iteration															
0	1	2	3	4	5	6	7	8	9	10	11	12	13		
there	there	there	there	there	there	there	on	there	there	though	by	on	on	that	that
on	on	on	on	on	on	on	there	on	on	on	on	by	by	on	on
that	that	that	though	that	by	by	though	though	though	there	such	such	such	by	by
who	until	what	that	what	what	what	what	by	until	by	though	until	though	such	such
until	what	though	what	though	that	until	by	what	what	what	almost	almost	until	until	until
any	who	until	by	until	until	same	until	until	by	then	almost	that	almost	until	almost
though	any	any	almost	by	though	almost	almost	around	that	such	what	what	that	almost	almost
what	though	by	until	almost	almost	around	then	that	then	almost	until	there	what	what	there
same	almost	almost	around	such	around	though	;	almost	around	until	that	then	then	then	then
almost	by	who	such	same	any	that	that	then	same	that	there		there	though	though
;	same	around	any	around	such	;	around	same	almost	same	then				
then	;	same	who	any	then	any	same	;	such	around	same				
by	then	;	same	;	;	then	any	such	;						
such	around	such	;	then	same	who	such	any							
around	such	then	then	who	who	such	who								
but	but	but	but	before	but	but									
before	how	how	before	but	before										
how	before	before	whether	whether											
whether	whether	whether	how												
within	within	within													
after	after														
,															

From the comparison of both tables with lists of attributes it is apparent that the rates of occurrence for the same attributes for both datasets were often significantly different. For example both punctuation marks for female writers withstood many iteration steps, in fact to the very last one, while for male authors they both disappeared after the 9th iteration. On the other hand, the attribute "that" for male writers lasted till the end, whereas for female dataset it was rejected after the initial run. Such differences were only natural, as importance of features was always considered in the local context and reflected data dependencies in a particular decision table.

Iteratively repeated execution of the procedure of reduct generation resulted not only in additional observation on characteristic of features, but also in assigning a specific score to all reducts, named a reduct survival index. RedSI for a reduct gives a number of iterations in which this reduct was found. RedSI equal zero means that the reduct was generated only for the original decision table, before any of the attributes were removed from the table. Each iteration involved discarding one variable, so RedSI as high as 15 indicates a reduct lasting through 15 iterations—and that despite many rejected attributes the reduct remained a reduct of the table. The numbers of reducts calculated at each iteration, and the numbers of reducts with the corresponding survival index are provided in Table 6. As can be expected, the numbers of reducts generated decreased with each iteration, with a single reducts found for the last one. In the case of reducts with specific RedSI, the trends were not strictly monotonic.

Table 6. Numbers of reducts generated by exhaustive algorithm in each iteration and the numbers of reducts with the corresponding survival index

	Female authors		Male authors			Female authors		Male authors	
Iteration	All	RedSI	All	RedSI	Iteration	All	RedSI	All	RedSI
0	914	57	12163	3278	8	58	24	192	101
1	857	158	8885	3025	9	34	11	91	50
2	699	198	5860	2077	10	23	5	41	21
3	501	148	3783	1521	11	18	8	20	12
4	353	109	2262	924	12	10	2	8	2
5	244	83	1338	596	13	4	4	3	
6	161	64	742	335	14	3-1	1	1	6
7	97	39	407	215	15	1	3		

The discovered reduct survival index was used as a way to grouping reducts from the extended representative sample, containing all elements found by genetic algorithm with some added reducts from the exhaustive search. This perspective, imposed on evaluation of performance for reduct-based rule classifiers, led to charts included in Fig. 3. It can be observed that low cardinality of reducts not necessarily resulted in their corresponding high survival indices. In fact, many short reduct did not survive beyond the first modification of the decision table, in particular for male writer dataset. Of particular importance is the conclusion that with increasing iterations, the diversity of classification outcome, the range of obtained values for classification accuracy became smaller and smaller, which was also confirmed by calculating again the averaged performance and standard deviation for samples, given in Table 7.

The columns of Table 7 list the results obtained for all reducts from the representative sample generated at a certain iteration step (column "All"), and limited to the group of reducts with specific RedSI. In the former case in calculations there were included also reducts with higher RedSI. The difference between the two approaches is clearly visible for 13th iteration for male writers: as there were no reducts with such RedSI, no average can be given for this column, but since there were some reducts with RedSI higher than 13, the average for all generated reducts for this step could be calculated, and it was the same as the one obtained as RedSI in the next iteration because it was the last one.

The increasing similarity of predictive powers for reduct-based classifiers obtained for reducts generated through iterative procedure, in particular for higher values of survival index, confirms the merit of the proposed methodology and shows that it can be used to support more informed choice of a reduct for classification model construction.

5. Conclusions

Reducts constitute an important notion and mechanism for feature reduction, as they preserve the predictive power of an information system by selected subset of attributes. From the perspective of data and knowledge representation the smaller a reduct the better, yet such considerations are insufficient when reducts are used for building classification

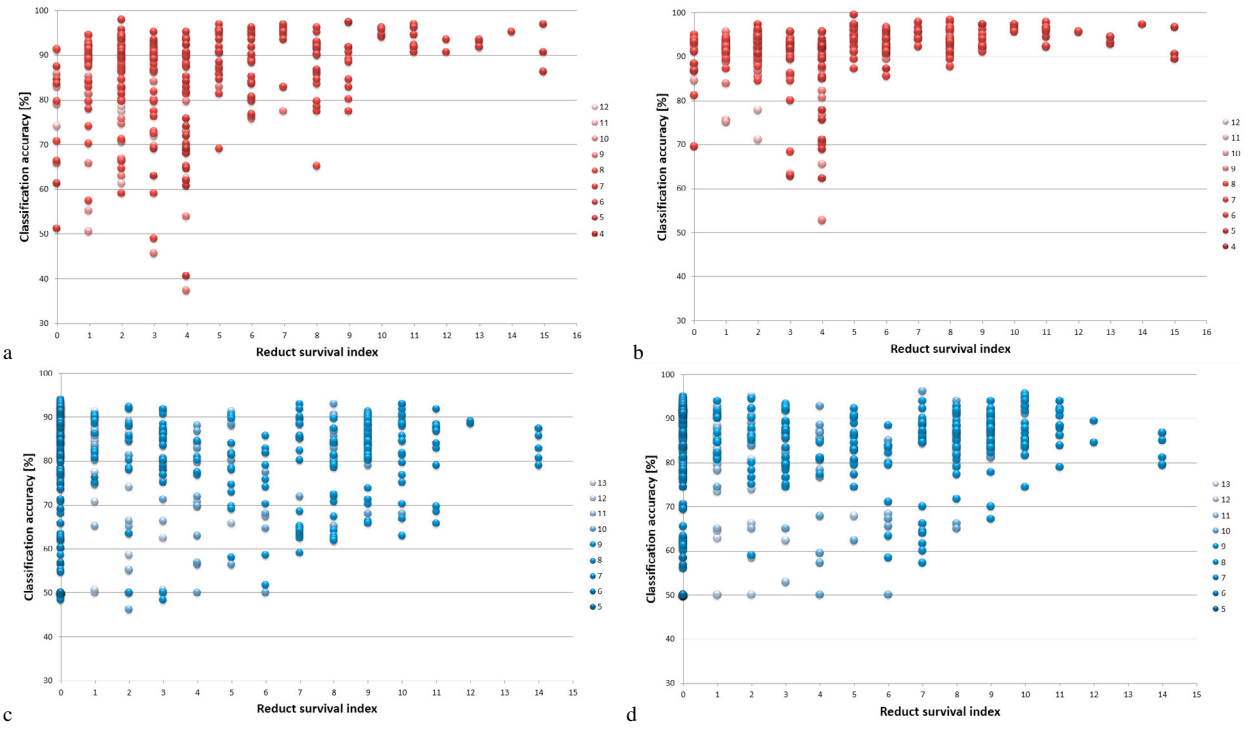

Figure 3. Performance [%] of selected reduct-based rule classifiers for: (a) and (b) female authors, (c) and (d) male authors, for (a) and (c) simple voting, and (b) and (d) standard voting strategy in case of conflicts.

Table 7. Averaged performance [%] and standard deviation for rule classifiers based on a representative sample of reducts generated by exhaustive algorithm per iteration (with specific survival index)

Iter.	Female authors				Male authors			
	Simple voting		Standard voting		Simple voting		Standard voting	
	All	RedSI	All	RedSI	All	RedSI	All	RedSI
0	84.10 ± 12.05	71.55 ± 17.32	90.95 ± 7.06	86.37 ± 11.10	78.13 ± 14.64	75.89 ± 16.73	81.46 ± 13.73	78.80 ± 16.50
1	84.48 ± 11.88	81.05 ± 14.14	91.02 ± 7.03	89.78 ± 06.04	79.57 ± 12.83	82.06 ± 10.97	83.09 ± 11.36	81.93 ± 11.77
2	84.63 ± 11.76	79.65 ± 15.14	91.14 ± 7.28	89.80 ± 06.85	79.21 ± 12.98	74.10 ± 16.66	83.26 ± 11.30	80.04 ± 14.35
3	85.04 ± 11.97	81.30 ± 13.01	91.05 ± 7.90	88.70 ± 08.55	79.81 ± 12.30	78.82 ± 14.08	83.64 ± 10.77	82.44 ± 11.32
4	85.92 ± 11.56	77.43 ± 15.89	91.60 ± 7.67	85.45 ± 11.60	79.65 ± 11.86	74.41 ± 14.38	83.59 ± 10.60	78.33 ± 13.54
5	89.12 ± 07.20	88.75 ± 07.65	93.92 ± 3.29	94.36 ± 03.30	80.19 ± 11.39	78.65 ± 12.60	84.12 ± 10.09	82.74 ± 10.22
6	89.21 ± 07.11	86.67 ± 07.74	93.83 ± 3.29	92.38 ± 04.16	80.39 ± 11.20	71.11 ± 14.54	84.30 ± 10.09	74.00 ± 15.03
7	90.12 ± 06.68	90.43 ± 07.67	94.35 ± 2.76	95.06 ± 01.97	81.27 ± 10.23	78.09 ± 13.43	85.28 ± 08.90	80.31 ± 13.72
8	90.07 ± 06.58	87.85 ± 07.66	94.23 ± 2.85	93.73 ± 03.12	81.87 ± 09.45	79.30 ± 10.35	86.21 ± 07.34	85.04 ± 08.73
9	91.63 ± 05.21	88.18 ± 07.03	94.59 ± 2.50	93.79 ± 02.52	83.05 ± 08.78	83.48 ± 07.58	86.75 ± 06.59	86.45 ± 06.50
10	93.29 ± 03.01	94.89 ± 01.03	94.98 ± 2.40	96.56 ± 01.00	82.53 ± 10.08	82.22 ± 10.90	87.13 ± 06.54	87.62 ± 06.96
11	92.84 ± 03.16	93.68 ± 02.59	94.54 ± 2.50	95.21 ± 02.16	82.86 ± 09.30	81.48 ± 11.23	86.61 ± 06.16	88.34 ± 04.99
12	92.17 ± 03.47	91.95 ± 01.97	94.00 ± 2.69	95.56 ± 00.78	84.93 ± 05.04	88.61 ± 02.75	84.03 ± 05.03	86.95 ± 03.54
13	92.22 ± 03.58	92.36 ± 01.00	93.61 ± 2.89	93.75 ± 00.99	84.93 ± 05.04		83.06 ± 05.15	
14	92.08 ± 05.02	95.00 ± 00.79	93.47 ± 04.13	97.23 ± 02.36		83.70 ± 05.06		83.06 ± 05.15
15		91.11 ± 05.67		92.22 ± 04.02				

models. Not only higher cardinality of reducts can lead to better performance of constructed classifiers, but even in the case of the same reduct length noticeable difference in predictions can be observed.

The reported research works focused on generation of reducts by several approaches, with the aim at supporting reduct selection, by providing some information on the similarity of considered candidates, from the point of view

of predictions by systems build on them. The proposed methodology of iterative reduct generation relies on keeping attributes occurring most often in reducts, and discarding these that appear rarely. For each reduct a survival index is defined by the number of iterations it withstood, corresponding to the number of modifications of the decision table.

The experiments included a search of reducts with the genetic algorithm, and the group found was further extended by selected cases obtained by exhaustive search. Such sample was used to illustrate the merit of the iterative reduct generation procedure, which resulted in noticeable decrease in the diversity of predictive properties, helping to detect reducts that could be considered as to be of the similar quality, from the perspective of classification.

In the future research other algorithms for finding reducts will be tested, such as greedy heuristics focused on optimisation of length.

Acknowledgements

The research works described in the paper were performed at the Silesian University of Technology, Gliwice, within the statutory project (RAU-6, 2020), and at the University of Silesia in Katowice, Institute of Computer Science.

References

[1] Alsolami, F., Amin, T., Moshkov, M., Zielosko, B., Żabiński, K., 2019. Comparison of heuristics for optimization of association rules. Fundamenta Informaticae 166, 1–14.

[2] Bazan, J., Skowron, A., Ślęzak, D., Wróblewski, J., 2003. Searching for the complex decision reducts: The case study of the survival analysis, in: Zhong, N., Raś, Z.W., Tsumoto, S., Suzuki, E. (Eds.), Foundations of Intelligent Systems, Springer Berlin Heidelberg. pp. 160–168.

[3] Bazan, J., Szczuka, M., 2005. The rough set exploration system, in: Peters, J.F., Skowron, A. (Eds.), Transactions on Rough Sets III. Springer, Berlin, Heidelberg. volume 3400 of *Lecture Notes in Computer Science*, pp. 37–56.

[4] Bazan, J.G., Skowron, A., Synak, P., 1994. Dynamic reducts as a tool for extracting laws from decisions tables, in: Raś, Z.W., Zemankova, M. (Eds.), Methodologies for Intelligent Systems, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 346–355.

[5] Chen, Y., Zhu, Q., Xu, H., 2015. Finding rough set reducts with fish swarm algorithm. Knowowledge-Based Systems 81, 22–29.

[6] Fayyad, U., Irani, K., 1993. Multi-interval discretization of continuous valued attributes for classification learning, in: Proceedings of the 13th International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers. pp. 1022–1027.

[7] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I., 2009. The WEKA data mining software: An update. SIGKDD Explorations 11, 10–18.

[8] Jensen, R., Shen, Q., 2003. Finding rough set reducts with ant colony optimization, in: Proceedings of the 2003 UK Workshop on Computational Intelligence, pp. 15–22.

[9] Koppel, M., Schler, J., Argamon, S., 2009. Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology 60, 9–26.

[10] Moshkov, M.J., Piliszczuk, M., Zielosko, B., 2007. On construction of partial reducts and irreducible partial decision rules. Fundamenta Informaticae 75, 357–374.

[11] Moshkov, M.J., Piliszczuk, M., Zielosko, B., 2008. Partial Covers, Reducts and Decision Rules in Rough Sets - Theory and Applications. volume 145 of *Studies in Computational Intelligence*. Springer.

[12] Pawlak, Z., Skowron, A., 2007a. Rough sets and boolean reasoning. Information Sciences 177, 41–73.

[13] Pawlak, Z., Skowron, A., 2007b. Rudiments of rough sets. Information Sciences 177, 3–27.

[14] Skowron, A., Rauszer, C., 1992. The discernibility matrices and functions in information systems, in: Słowiński, R. (Ed.), Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory. Springer Netherlands, Dordrecht, pp. 331–362.

[15] Stamatatos, E., 2009. A survey of modern authorship attribution methods. Journal of the Americal Society for Information Science and Technology 60, 538–556.

[16] Stańczyk, U., 2013. Weighting of attributes in an embedded rough approach, in: Gruca, A., Czachórski, T., Kozielski, S. (Eds.), Man-Machine Interactions 3. Springer-Verlag, Berlin, Germany. volume 242 of *AISC*, pp. 475–483.

[17] Stańczyk, U., Zielosko, B., 2019. On approaches to discretisation of stylometric data and conflict resolution in decision making, in: Rudas, I.J., Csirik, J., Toro, C., Botzheim, J., Howlett, R.J., Jain, L.C. (Eds.), Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES-2019, Budapest, Hungary, 4-6 September 2019. Elsevier. volume 159 of *Procedia Computer Science*, pp. 1811–1820.

[18] Stańczyk, U., Zielosko, B., Żabiński, K., 2018. Application of greedy heuristics for feature characterisation and selection: A case study in stylometric domain, in: Nguyen, H., Ha, Q., Li, T., Przybyla-Kasperek, M. (Eds.), Proceedings of the International Joint Conference on Rough Sets, IJCRS 2018. Springer, Quy Nhon, Vietnam. volume 11103 of *Lecture Notes in Computer Science*, pp. 350–362.

[19] Stawicki, S., Ślęzak, D., Janusz, A., Widz, S., 2017. Decision bireducts and decision reducts - a comparison. Int. J. Approx. Reason. 84, 75–109.

[20] Wróblewski, J., 1996. Theoretical foundations of order-based genetic algorithms. Fundamenta Informaticae 28, 423–430.

[21] Wróblewski, J., 1998. Genetic algorithms in decomposition and classification problems, in: Polkowski, L., Skowron, A. (Eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems. Physica-Verlag HD, Heidelberg, pp. 471–487.