On Positive-Correlation-Promoting Reducts

Joanna Henzel[1], Andrzej Janusz[2], Marek Sikora[1], and Dominik Ślęzak[2](✉)

[1] Department of Computer Networks and Systems, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Gliwice, Poland
[2] Institute of Informatics, Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland
slezak@mimuw.edu.pl

Abstract. We introduce a new rough-set-inspired binary feature selection framework, whereby it is preferred to choose attributes which let us distinguish between objects (cases, rows, examples) having different decision values according to the following mechanism: for objects $u1$ and $u2$ with decision values $dec(u1) = 0$ and $dec(u2) = 1$, it is preferred to select attributes a such that $a(u1) = 0$ and $a(u2) = 1$, with the secondary option – if the first one is impossible – to select a such that $a(u1) = 1$ and $a(u2) = 0$. We discuss the background for this approach, originally inspired by the needs of the genetic data analysis. We show how to derive the sets of such attributes – called positive-correlation-promoting reducts (PCP reducts in short) – using standard calculations over appropriately modified rough-set-based discernibility matrices. The proposed framework is implemented within the RoughSets R package which is widely used for the data exploration and knowledge discovery purposes.

Keywords: Rough sets · Feature selection · Discernibility · Rule induction · Positive-correlation-promoting reducts · RoughSets R package

1 Introduction

Rough set approaches are successfully utilized in the areas of machine learning and knowledge discovery, particularly for feature selection and classifiers simplification, as well as for deriving easily interpretable decision models from the data [1,9]. There are a number of generalizations and hybridizations of rough set methods available in the form of software toolkits, including dominance-based rough set algorithms [3], fuzzy-rough set algorithms [10], and others. There is plenty of research connecting rough sets with other knowledge representation methodologies such as e.g. formal concept analysis [4], as well as application-oriented studies such as e.g. extensions of standard rough set techniques aimed at handling high-dimensional data sets [7]. Finally, it is worth noting that rough set approaches can be combined in a natural way with various symbolic machine learning methods, in particular those designed for rule induction [5,12].

R. Bello et al. (Eds.): IJCRS 2020, LNAI 12179, pp. 213–221, 2020.
https://doi.org/10.1007/978-3-030-52705-1_16

```
library(RoughSets)
```

```
## Loading required package: Rcpp
```

```
data(RoughSetData)
binary.dt <- RoughSetData$binary.dt

# our exemplary decision table
binary.dt
```

V1 <fctr>	V2 <fctr>	V3 <fctr>	V4 <fctr>	V5 <fctr>	V6 <fctr>	V7 <fctr>	V8 <fctr>	V9 <fctr>	V10 <fctr>	dec <fctr>
1	0	1	1	1	1	0	0	1	1	0
1	0	0	1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1	0	1	0
1	0	1	0	0	1	0	1	1	0	0
0	1	1	1	1	0	1	1	0	1	0
1	0	1	0	0	0	1	1	0	1	1
0	1	0	1	0	0	1	1	0	0	1
0	1	0	1	0	1	0	0	1	1	1
0	1	1	0	1	0	1	0	1	0	1
0	1	1	1	1	1	0	0	1	0	1

1-10 of 10 rows

Fig. 1. Example of a binary decision table with 10 objects, 10 attributes $V1, ..., V10$, as well as decision *dec*, displayed using the RoughSets R package [10].

In this study, we are interested in inducing rules which follow a specific pattern of selecting conditions pointing at particular decisions. Using an example of decision table in Fig. 1, we seek for rules describing the case $dec = 1$ with conditions $Vi = 1$ (e.g. $V2 = 1 \wedge V6 = 1 \Rightarrow dec = 1$ supported by rows 8 and 10) and $dec = 0$ with conditions $Vi = 0$ (e.g. $V8 = 0 \wedge V9 = 0 \Rightarrow dec = 0$ supported by row 2). Only if there is no other choice, we would allow additional descriptors of the form $Vi = 0$ for $dec = 1$ and $Vi = 1$ for $dec = 0$ (e.g. it is impossible to construct a rule covering row 7 without using conditions $Vi = 0$).

We propose a new approach to feature selection, aimed at finding attributes which are suitable for constructing such rules. In order to do this, we modify the rough-set-based notion of a reduct [9]. For the binary data, our *positive-correlation-promoting* (*PCP*) *reducts* will prefer to contain attributes Vi such that – for objects $u1$ and $u2$ with decisions $dec(u1) = 0$ and $dec(u2) = 1$ – there is $Vi(u1) = 0$ and $Vi(u2) = 1$, or else – but only if the former option does not hold for any attribute – there is $Vi(u1) = 1$ and $Vi(u2) = 0$. (On the contrary, both those options of discernibility – i.e. $Vi(u1) = 0, Vi(u2) = 1$ versus $Vi(u1) = 1, Vi(u2) = 0$ – have the same importance for standard reducts.)

Going further, in Sect. 2 we recall the RoughSets R package [10], whereby we implement our new approach. In Sect. 3 we present the background for PCP reducts. In Sect. 4 we show that they are derivable using a modification of

```
## building a classical decision-relative discernibility matrix
disc.matrix <- BC.discernibility.mat.RST(binary.dt, return.matrix = TRUE)
head(disc.matrix$disc.list)
```

```
## [[1]]
## [1] "V4" "V5" "V6" "V7" "V8" "V9"
##
## [[2]]
## [1] "V1"  "V2"  "V3"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10"
##
## [[3]]
## [1] "V1" "V2" "V3" "V5"
##
## [[4]]
## [1] "V1"  "V2"  "V4"  "V6"  "V7"  "V10"
##
## [[5]]
## [1] "V1"  "V2"  "V10"
##
## [[6]]
## [1] "V3"  "V4"  "V7"  "V8"  "V10"
```

	1	2	3	4	5
6	V4, V5, V6, V7, V8, V9	V3, V4, V7, V8, V10	V4, V6	V6, V7, V9, V10	V1, V2, V4, V5
7	V1, V2, V3, V5, V6, V7, V8, V9, V10	V1, V2, V7, V8	V1, V2, V3, V6, V10	V1, V2, V3, V4, V6, V7, V9	V3, V5, V10
8	V1, V2, V3, V5	V1, V2, V6, V9, V10	V1, V2, V3, V7, V8, V9	V1, V2, V3, V4, V8, V10	V3, V5, V6, V7, V8, V9
9	V1, V2, V4, V6, V7, V10	V1, V2, V3, V4, V5, V7, V9	V1, V2, V4, V5, V6, V8, V9, V10	V1, V2, V5, V6, V7, V8	V4, V8, V9, V10
10	V1, V2, V10	V1, V2, V3, V5, V6, V9	V1, V2, V5, V7, V8, V9, V10	V1, V2, V4, V5, V8	V6, V7, V8, V9, V10

Fig. 2. Example continued: [Top] An excerpt from the standard discernibility matrix computed for decision table in Fig. 1 using the RoughSets R package; [Bottom] Full standard discernibility matrix for the considered decision table.

rough-set-based discernibility matrices. In Sect. 5 we discuss future relevant extensions of the package. In Sect. 6 we conclude the paper.

2 About the RoughSets R Package

The RoughSets package is available in CRAN (http://cran.r-project.org/web/packages/RoughSets/index.html). Its newest version can be found also in GitHub (https://github.com/janusza/RoughSets). It provides implementations of classical rough-set-based methods and their fuzzy-related extensions for data modeling and analysis. In particular, it includes tools for feature selection and attribute reduction, as well as rule induction and rule-based classification.

Figures 2 and 3 illustrate two out of the most fundamental functionalities of the package – calculation of a discernibility matrix from the input decision table, and calculation of all decision reducts from the input discernibility matrix. Let us recall that decision tables stand for standard representation of the labeled tabular data in the rough set framework. Discernibility matrices assign the pairs of objects (rows) having different decision values with attributes which are able to distinguish between them. Decision reducts are irreducible attribute subsets which distinguish between all such pairs (for decision table in Fig. 1, one needs to distinguish rows 1, 2, 3, 4, 5 from 6, 7, 8, 9, 10), i.e., those which have non-empty intersection with every cell of the corresponding matrix.

```r
# computation of all classical reducts
classic.reducts <- FS.all.reducts.computation(disc.matrix)

# top 3 classical reducts
head(classic.reducts$decision.reduct, 3)
```

```
## $reduct1
## A feature subset consisting of 3  attributes:
## V1, V6, V10
##
## $reduct2
## A feature subset consisting of 3  attributes:
## V2, V6, V10
##
## $reduct3
## A feature subset consisting of 4  attributes:
## V1, V3, V4, V10
```

```r
# a total number of found reducts
cat("A total number of reducts found: ",
    length(classic.reducts$decision.reduct), "\n", sep = "")
```

```
## A total number of reducts found: 38
```

```r
# a decision core of the data table in empty...
classic.reducts$core
```

```
## character(0)
```

Fig. 3. Example continued: Standard reducts for decision table displayed in Fig. 1, calculated using the all-reducts function in the RoughSets R package.

The considered package contains also other, more modern methods of decision reduct calculation. Some of them work on far more efficient data structures than discernibility matrices. Some of them search heuristically for single reducts or small groups of reducts instead of all of them. Nevertheless, referring to functions in Figs. 2 and 3 is a good starting point for further investigations.

3 Inspiration for PCP Reducts

The idea of operating with rules exemplified in Sect. 1 comes from our earlier studies on the data produced in the cancer genome atlas project (https://en.wiki\discretionary-pedia.org/wiki/The_Cancer_Genome_Atlas) [11] and other gene-related data sets [7]. Let us consider the copy number variation pipeline (https://en.wikipedia.org/wiki/Copy-number_variation) [6] which uses the Affymetrix SNP 6.0 array data [2] to identify the repeating genomic regions and to infer the copy number of those repeats. Imagine that attributes in Fig. 1 represent some of the protein coding genes and rows represent patient samples. For each patient, a gene can be characterized by 0 (no change) or 1 (change in the copy number for that gene). Assume that *dec* takes value 1 for patients with short survival time. Then, we would like to describe decision class $dec = 1$ by genes for which a change in the copy number was registered i.e. using conditions $Vi = 1$.

```
## building a PCP discernibility matrix
disc.pcp.matrix <- BC.discernibility.positive.mat.RST(binary.dt, return.matrix = TRUE)
head(disc.pcp.matrix$disc.list)
```

```
## [[1]]
## [1] "V7" "V8"
##
## [[2]]
## [1] "V3"  "V7"  "V8"  "V10"
##
## [[3]]
## [1] "V4" "V6"
##
## [[4]]
## [1] "V7"  "V10"
##
## [[5]]
## [1] "V1"
##
## [[6]]
## [1] "V2" "V7" "V8"
```

	1	2	3	4	5
6	V7, V8	V3, V7, V8, V10	V4, V6	V7, V10	V1
7	V2, V7, V8	V2, V7, V8	V2	V2, V4, V7	V3, V5, V10
8	V2	V2, V6, V9, V10	V2, V9	V2, V4, V10	V6, V9
9	V2, V7	V2, V3, V5, V7, V9	V2, V5, V9	V2, V5, V7	V9
10	V2	V2, V3, V5, V6, V9	V2, V5, V9	V2, V4, V5	V6, V9

Fig. 4. Example continued: [Top] An excerpt from the PCP discernibility matrix computed for decision table in Fig. 1 using the new functionality of the RoughSets R package; [Bottom] Full PCP discernibility matrix for the considered table.

If we were interested only in such rules, then they could be modeled using formal concept analysis [4]. However, we also need rules describing $dec = 0$ by $Vi = 0$. In such a case, one might suggest that it is worth using the dominance-based rough set framework [3]. However, it is not so strict that we should use only conditions $Vi = 1$ for $dec = 1$ and $Vi = 0$ for $dec = 0$. Such conditions are preferred and should be *promoted* by the rule generation process. However, if it is impossible to form rules using only such conditions, then the other ones ($Vi = 0$ for $dec = 1$ and $Vi = 1$ for $dec = 0$) are allowed too.

4 Discernibility Characteristics of PCP Reducts

Discussion in the previous section leads us toward the following:

Definition 1. *Let a binary decision table $\mathbb{A} = (U, A \cup \{d\})$ be given. (In Fig. 1: $U = \{u1, ..., u10\}$, $A = \{V1, ..., V10\}$, $d = dec$.) Consider object $u \in U$ and subset $B \subseteq A$. We say that rule $\bigwedge_{a \in B} a = a(u) \Rightarrow d = d(u)$ is a positive-correlation-promoting (PCP) rule, if and only if it holds irreducibly in $\mathbb{A}$ and there is $\forall_{a \in B}\, a(u) = d(u)$ or else, one cannot replace conditions $a = a(u)$ such that $a(u) \neq d(u)$ with any conditions $b = b(u)$, $b(u) = d(u)$, $b \notin B$.*

Definition 2. *Subset $B \subseteq A$ is a positive-correlation-promoting (PCP) reduct, if and only if each $u \in U$ can be covered by a PCP rule $\bigwedge_{a \in B^u} a = a(u) \Rightarrow d = d(u)$, $B^u \subseteq B$, and there is no proper $B' \subsetneq B$ with this property.*

```
# analogically, computation of all PCP reducts
pcp.reducts <- FS.all.reducts.computation(disc.pcp.matrix)

# top 3 PCP reducts - notice how large they are relative to the classical ones
head(pcp.reducts$decision.reduct, 3)
```

```
## $reduct1
## A feature subset consisting of 6  attributes:
## V1, V2, V3, V4, V7, V9
##
## $reduct2
## A feature subset consisting of 6  attributes:
## V1, V2, V3, V6, V7, V9
##
## $reduct3
## A feature subset consisting of 6  attributes:
## V1, V2, V4, V5, V7, V9
```

```
# a total number of reducts found
cat("A total number of PCP reducts found: ",
    length(pcp.reducts$decision.reduct), "\n", sep = "")
```

```
## A total number of PCP reducts found: 8
```

```
# a decision core for PCP reducts contains attributes V1 and V2,
# even though they provide the same information in a classical sense:
pcp.reducts$core
```

```
## [1] "V1" "V2" "V9"
```

Fig. 5. Example continued: The all-reducts function in the RoughSets R package, now executed on the PCP discernibility matrix for decision table in Fig. 1.

The following characteristics can be shown in straightforward way:

Proposition 1. *For binary* $\mathbb{A} = (U, A \cup \{d\})$*, for every* $u1, u2 \in U$ *such that* $d(u1) \neq d(u2)$*, define* $M^{+}(u1, u2) = \{a \in A : a(u1) = d(u1) \wedge a(u2) = d(u2)\}$ *and* $M^{-}(u1, u2) = \{a \in A : a(u1) \neq d(u1) \wedge a(u2) \neq d(u2)\}$*. Consider the PCP discernibility matrix which labels the pairs of objects as follows:*

$$M(u1, u2) = \begin{cases} M^{+}(u1, u2) \text{ if } M^{+}(u1, u2) \neq \emptyset \\ M^{-}(u1, u2) \text{ otherwise} \end{cases} \tag{1}$$

Then, a given $B \subseteq A$ *is a PCP reduct, if and only if it is an irreducible subset such that* $B \cap M(u1, u2) \neq \emptyset$ *for every* $u1, u2 \in U$*,* $d(u1) \neq d(u2)$*.*

5 Heuristic Search of PCP Reducts and Rules

Definition 2 reflects the requirements of the feature selection process if the ultimate goal is to induce the rules of the form discussed in previous sections and considered earlier in [11]. Moreover, Proposition 1 provides us with an easy way to derive PCP reducts. Namely, it is enough to modify classical discernibility matrices and then apply the same techniques as those outlined in [8].

```
# decision tables reduced based on the top 2 classical reducts and one PCP reduct
dt.classic1 = SF.applyDecTable(binary.dt, classic.reducts$decision.reduct[[1]])
dt.classic2 = SF.applyDecTable(binary.dt, classic.reducts$decision.reduct[[2]])
dt.pcp = SF.applyDecTable(binary.dt, pcp.reducts$decision.reduct[[1]])

# computing decision rules using the LEM2 algorithm
rules.classic1 <- RI.LEM2Rules.RST(dt.classic1)
rules.classic1
```

```
## A set consisting of  6  rules:
## 1. IF V1 is 1 and V6 is 1 THEN dec is 0;
##      (supportSize=3; laplace=0.8)
## 2. IF V6 is 0 and V1 is 0 and V10 is 1 THEN dec is 0;
##      (supportSize=1; laplace=0.6667)
## 3. IF V1 is 1 and V10 is 0 THEN dec is 0;
##      (supportSize=2; laplace=0.75)
## 4. IF V1 is 0 and V10 is 0 THEN dec is 1;
##      (supportSize=3; laplace=0.8)
## 5. IF V1 is 0 and V6 is 1 THEN dec is 1;
##      (supportSize=2; laplace=0.75)
## 6. IF V1 is 1 and V6 is 0 and V10 is 1 THEN dec is 1;
##      (supportSize=1; laplace=0.6667)
```

```
rules.classic2 <- RI.LEM2Rules.RST(dt.classic2)
rules.classic2
```

```
## A set consisting of  6  rules:
## 1. IF V2 is 0 and V6 is 1 THEN dec is 0;
##      (supportSize=3; laplace=0.8)
## 2. IF V2 is 0 and V10 is 0 THEN dec is 0;
##      (supportSize=2; laplace=0.75)
## 3. IF V2 is 1 and V6 is 0 and V10 is 1 THEN dec is 0;
##      (supportSize=1; laplace=0.6667)
## 4. IF V2 is 1 and V10 is 0 THEN dec is 1;
##      (supportSize=3; laplace=0.8)
## 5. IF V10 is 1 and V2 is 0 and V6 is 0 THEN dec is 1;
##      (supportSize=1; laplace=0.6667)
## 6. IF V2 is 1 and V6 is 1 THEN dec is 1;
##      (supportSize=2; laplace=0.75)
```

```
rules.pcp <- RI.LEM2Rules.RST(dt.pcp)
rules.pcp
```

```
## A set consisting of  5  rules:
## 1. IF V2 is 0 and V7 is 0 THEN dec is 0;
##      (supportSize=3; laplace=0.8)
## 2. IF V9 is 0 and V3 is 1 and V4 is 1 THEN dec is 0;
##      (supportSize=2; laplace=0.75)
## 3. IF V2 is 1 and V9 is 1 THEN dec is 1;
##      (supportSize=3; laplace=0.8)
## 4. IF V2 is 1 and V3 is 0 THEN dec is 1;
##      (supportSize=2; laplace=0.75)
## 5. IF V4 is 0 and V7 is 1 THEN dec is 1;
##      (supportSize=2; laplace=0.75)
```

Fig. 6. Decision rules derived using the LEM2 [5] algorithm's version available in the RoughSets R package [10]. The rules are derived for two examples of standard reducts and one example of a PCP reduct (the last one). This means that only attributes contained in the given reduct are considered as input to LEM2. Although PCP reducts are designed to promote attributes which let us construct rules including more descriptors of the form $Vi = 1$ pointing at decision $dec = 1$, as well as more descriptors of the form $Vi = 0$ pointing at $dec = 0$, this information is lost during the phase of rule shortening. This is because – in its current implementation – this phase does not distinguish between positively (M^+) and negatively (M^-) correlated discernibility cases.

This fact allowed us to extend the RoughSets package [10], as visible in Figs. 4 and 5. When comparing the PCP matrix (Fig. 4) with its classical counterpart (Fig. 2), one can see that the attribute sets are now smaller. (The only unchanged cells are $M(u3, u6)$ and $M(u5, u7)$ – this is because $M^+ = \emptyset$ in both cases.) Consequently, PCP reducts are bigger than standard ones. In particular, PCP reducts can include both attributes $V1$ and $V2$ which are mutually interchangeable [7], so they would never co-occur in a standard reduct.

Still, there is a lot left to be done in the area of heuristic extraction of PCP reducts. One might expect that the corresponding algorithms should seek for PCP reducts which yield rules with maximum number of descriptors $Vi = 1$ for $dec = 1$ (and $Vi = 0$ for $dec = 0$). Unfortunately, classical methods cannot distinguish between the cases $M^+ \neq \emptyset$ and $M^+ = \emptyset$ in equation (1), so their heuristic optimization functions do not work properly. The same happens with standard rule induction methods [5,10] as further outlined in Fig. 6.

6 Further Research Directions

The newly introduced PCP reducts require further study in many aspects. Besides the aforementioned need of better heuristic search methods, we shall design algorithms working on more efficient data structures than PCP matrices. Herein, we will attempt to adapt some of modern data structures which are used to derive classical reducts and rules in rough-set-based toolkits [3,5].

Another future direction may refer to PCP reducts for non-binary data sets. In this paper, the nature of *promoting positive correlations* was expressed in terms of selecting these attributes which share – if possible – the same value differences as observed for the decision column. An analogous idea could be considered e.g. for numerical data sets, whereby one may think about appropriate modifications of fuzzy-rough discernibility characteristics [4,10].

References

1. Bello, R., Falcon, R.: Rough sets in machine learning: a review. In: Wang, G., Skowron, A., Yao, Y., Ślęzak, D., Polkowski, L. (eds.) Thriving Rough Sets. SCI, vol. 708, pp. 87–118. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54966-8_5
2. Bhaskar, H., Hoyle, D.C., Singh, S.: Machine learning in bioinformatics: a brief survey and recommendations for practitioners. Comp. Bio. Med. **36**(10), 1104–1125 (2006)
3. Błaszczyński, J., Greco, S., Matarazzo, B., Słowiński, R., Szeląg, M.: jMAF - dominance-based rough set data analysis framework. In: Skowron, A., Suraj, Z. (eds.) Rough Sets and Intelligent Systems, vol. 1, pp. 185–209. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30344-9_5
4. Cornejo Piñero, M.E., Medina-Moreno, J., Ramírez-Poussa, E.: Fuzzy-attributes and a method to reduce concept lattices. In: Cornelis, C., et al. (eds.) RSCTC 2014. LNCS (LNAI), vol. 8536, pp. 189–200. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08644-6_20

5. Grzymała-Busse, J.W.: A comparison of rule induction using feature selection and the LEM2 algorithm. In: Stańczyk, U., Jain, L.C. (eds.) Feature Selection for Data and Pattern Recognition. SCI, vol. 584, pp. 163–176. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-45620-0_8
6. Jakobsson, M., et al.: Genotype, haplotype and copy-number variation in worldwide human populations. Nature **451**(7181), 998–1003 (2008)
7. Janusz, A., Ślęzak, D.: Rough set methods for attribute clustering and selection. Appl. Artif. Intell. **28**(3), 220–242 (2014)
8. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Inf. Sci. **177**(1), 41–73 (2007)
9. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. **177**(1), 3–27 (2007)
10. Riza, L.S., Janusz, A., Bergmeir, C., Cornelis, C., Herrera, F., Ślęzak, D., Benítez, J.M.: Implementing algorithms of rough set theory and fuzzy rough set theory in the R package, "RoughSets". Inf. Sci. **287**, 68–89 (2014)
11. Sikora, M., Gruca, A.: Induction and selection of the most interesting gene ontology based multiattribute rules for descriptions of gene groups. Pattern Recognit. Lett. **32**(2), 258–269 (2011)
12. Sikora, M., Wróbel, L., Gudyś, A.: GuideR: a guided separate-and-conquer rule learning in classification, regression, and survival settings. Knowl. Based Syst. **173**, 1–14 (2019)