Representatives of Rough Regions for Generating Classification Rules

Piotr Hońko

Representatives of Rough Regions for Generating Classification Rules

Piotr Hońko *

Faculty of Computer Science, Bialystok University of Technology, Wiejska 45A, 15-351 Białystok, Poland
p.honko@pb.edu.pl

Abstract. Rough set theory provides a useful tool for describing uncertain concepts. The description of a given concept constructed based on rough regions can be used to improve the quality of classification. Processing large data using rough set methods requires efficient implementations as well as alternative approaches to speed up computations. This paper proposes a representative-based approach for rough region-based classification. Positive, boundary, and negative regions are replaced with their representatives sets that preserve information needed for generating classification rules. For data divisible into a relatively low number of equivalence classes representatives sets are considerably smaller than the whole regions. Using a small representation of regions significantly speeds up the process of rule generation.

Keywords: rough sets, classification rules, representative-based approach

1 Introduction

Rough set theory [5] has found a wide application in processing uncertain data. It enables to describe a concept represented by a subset of the universe in an alternative way. Such a description identifies objects that certainly (lower approximation) and possibly (upper approximation) belong to the concept. The thus created rough set (i.e. the pair of lower and upper approximations) can further be used to improve classification of uncertain data. For example, based on the lower (upper) approximation one can construct classification rules that assign objects to the concept certainly (possibly) correctly.

Processing large data using rough set methods may be a challenging task. Some methods require a more efficient implementation to make them applicable to processing huge amount of data. For example, a direct implementation of the definition of rough sets requires polynomial time to find approximations of a given concept. These computations can be speeded up e.g. by adapting sort algorithm that needs linear-logarithmic time [4].

Even though using efficient implementations, processing the whole data may still be time or space consuming. Another solution for shortening the run-time

* The project was funded by the National Science Center awarded on the basis of the decision number DEC-2012/07/B/ST6/01504.

and saving memory is to apply a representative-based approach. Instead of the whole data, only its previously found representatives are processed. The crucial step in this approach is to find representatives that preserve information essential for the goal of data processing.

Several representative-based approaches were applied in rough set theory. In [2] the set of representatives is defined as a minimal universe subset whose objects possess all properties under consideration. A similarity relation-based rough set approach was used in [3] to find representative cases in the problem of case-based reasoning. In [1] an object for which there is only one minimal universe subset (each subset belongs to a given covering of the universe) the object belongs to is understood as the representative of this subset. A minimal set of representatives in tolerance rough sets is identified in [6] by a prime implicant of the Boolean function over the variables corresponding to objects. Representatives used in [7] for the classification task are constructed based on maximal neighborhoods of objects in the positive region (i.e. lower approximation).

The goal of this paper is to propose a representatives sets of rough regions for classification rule generation. Positive, boundary, and negative regions (that can be constructed based on lower and upper approximations) are replaced with their representatives sets. A representative is defined as a pair of information vector corresponding to an object (limited to the attributes under consideration) and the cardinality of the equivalence class represented by the vector. The representatives sets of the three regions are constructed simultaneously using sort-based algorithm. Classification rules are generated from the representatives sets using an adaptation of sequential covering algorithm.

Unlike the above-described approaches, a representative proposed in this study includes not only an object of the universe but also information on its equivalence class that is essential for generating classification rules. Benefits of the approach are clearly visible for data that is divisible into relatively low number of equivalence classes. For such data representatives sets of regions are considerably smaller than the whole regions, thereby the generation of classification rules can significantly be accelerated.

The rest of the paper is organized as follows. Section 2 restates basic notion from rough set theory and proposes a representative-based approach for rough region-based classification rule generation. Section 3 develops algorithms using the proposed approach. Experimental research is reported in Section 4. Concluding remarks are provided in Section 5.

2 Rough Regions and Their Representatives for Generating Classification Rules

This section restates basic notions from rough set theory. It also proposes representatives sets of rough regions and shows how to use them in generation of classification rules.

2.1 Rough Regions

To store data to be processed, an information system is used.

Definition 1. *[5] (information system) An information system is a pair* $IS = (U, A)$*, where* U *is a non-empty finite set of objects, called the universe, and* A *is a non-empty finite set of attributes.*
Each attribute $a \in A$ *is treated as a function* $a : U \to V_a$*, where* V_a *is the value set of* a*.*

An object $x \in U$ can be represented by an information vector $\mathbf{x} = (a(x) : a \in A)$. Let $\mathbf{x}_B = (a(x) : a \in B)$, where $B \subseteq A$, be a B-information vector.

Essential information about data is expressed by an indiscernibility relation.

Definition 2. *[5] (indiscernibility relation) An indiscernibility relation* $IND(B)$ *generated by* $B \subseteq A$ *on* U *is defined by* $IND(B) = \{(x, y) \in U \times U : \forall_{a \in B} a(x) = a(y)\}$.

Any indiscernibility relation partitions the universe into equivalence classes. Let $\mathcal{E} = U/IND(B)$ be the family of equivalence classes of $IND(B)$ on U.

A concept that is not definable (i.e. it is not a union of some equivalence classes) can alternatively be described using approximations.

Definition 3. *[5] (rough approximations) Let* $AS_B = (U, IND(B))$ *be an approximation space. The lower and upper approximations of a subset* $X \subseteq U$ *in* AS *are defined, respectively, by*

1. $LOW(AS_B, X) = \bigcup\{E \in \mathcal{E} : E \subseteq X\}$,
2. $UPP(AS_B, X) = \bigcup\{E \in \mathcal{E} : E \cap X \neq \emptyset\}$.

Using approximations one can partition the universe into three regions.

Definition 4. *[5] (rough regions) The positive, boundary, and negative regions of a subset* $X \subseteq U$ *in* $AS_B = (U, IND(B))$ *are defined, respectively, by*

1. $POS(AS_B, X) = LOW(AS_B, X) = \bigcup\{E \in \mathcal{E} : E \subseteq X\}$,
2. $BND(AS_B, X) = UPP(AS_B, X) \setminus LOW(AS_B, X) = \bigcup\{E \in \mathcal{E} : E \cap X \neq \emptyset \wedge E \not\subseteq X\}$,
3. $NEG(AS_B, X) = U \setminus UPP(AS_B, X) = \bigcup\{E \in \mathcal{E} : E \cap X = \emptyset\}$.

The positive (negative) region includes objects that certainly belong (not belong) to the concept. The boundary region includes, in turn, uncertain objects.

2.2 Representatives of Rough Regions

The following representatives of equivalence classes and regions are proposed.

Let $\widehat{E}_B = (\mathbf{x}_B, c)$ be a representative of an equivalence class E, where $B \subseteq A, x \in E$ and $c = |E|$.

Definition 5. *(representatives sets of rough regions) The representatives sets of positive, boundary, and negative regions of a subset $X \subseteq U$ in $AS_B = (U, IND(B))$ are defined, respectively, by*

1. $\widehat{POS}(AS_B, X) = \{\widehat{E}_B : E \subseteq X\}$,
2. $\widehat{BND}(AS_B, X) = \{\widehat{E}_B : E \cap X \neq \emptyset \wedge E \not\subseteq X\}$,
3. $\widehat{NEG}(AS_B, X) = \{\widehat{E}_B : E \cap X = \emptyset\}$.

If the concept to be approximated is known before computing the equivalence classes, its rough regions can be defined using the equivalence classes computed separately for the concept and its complement.

Let $\mathcal{E}_X = X/IND(B)$, where $X \subseteq U$. Let also $\widehat{E}_B \simeq \widehat{E'}_B \Leftrightarrow \mathbf{x}_B = \mathbf{y}_B$, where $\widehat{E}_B = (\mathbf{x}_B, c)$ and $\widehat{E'}_B = (\mathbf{y}_B, c')$.

The following alternative definition of rough regions is proposed.

Proposition 1. *For any $X \subseteq U$ in $AS_B = (U, IND(B))$ the following hold.*

1. $POS(AS_B, X) = \bigcup\{E \in \mathcal{E}_X : \forall_{E' \in \mathcal{E}_{X^c}} \widehat{E}_B \not\simeq \widehat{E'}_B\}$;
2. $BND(AS_B, X) = \bigcup\{E \cup E' : E \in \mathcal{E}_X, E' \in \mathcal{E}_{X^c} : \widehat{E}_B \simeq \widehat{E'}_B\}$.
3. $NEG(AS_B, X) = \bigcup\{E \in \mathcal{E}_{X^c} : \forall_{E' \in \mathcal{E}_X} \widehat{E}_B \not\simeq \widehat{E'}_B\}$.

Proof. 1. Let $L = POS(AS_B, X)$ and $R = \bigcup\{E \in \mathcal{E}_X : \forall_{E' \in \mathcal{E}_{X^c}} \widehat{E}_B \not\simeq \widehat{E'}_B\}$.
The case "$\Rightarrow$". We have $E \subseteq X \Rightarrow X^c \subseteq U \setminus E$ (a), $\mathcal{E}_X = \{E \cap X : E \in \mathcal{E}\}$(b). We obtain $E \subseteq L \Leftrightarrow E \in \mathcal{E} \wedge E \subseteq X \Leftrightarrow \forall_{(x,y) \in E \times U \setminus E}(x, y) \notin IND(B) \wedge E \cap X = E \Rightarrow \forall_{(x,y) \in E \times X^c}(x, y) \notin IND(B) \wedge E \in \mathcal{E}_X$ by (a) and (b). Hence, we obtain $E \in \mathcal{E}_X \wedge \forall_{E' \in \mathcal{E}_{X^c}} \widehat{E}_B \not\simeq \widehat{E'}_B \Leftrightarrow E \subseteq R$.
The case "$\Leftarrow$". We have $E \in \mathcal{E}_X \Rightarrow E \subseteq X$(c), $E \in \mathcal{E}_X$
$\Leftrightarrow \forall_{(x,y) \in E \times X \setminus E}(x, y) \notin IND(B)$(d).
We obtain $E \subseteq R \Leftrightarrow E \in \mathcal{E}_X \wedge \forall_{E' \in \mathcal{E}_{X^c}} \widehat{E}_B \not\simeq \widehat{E'}_B \Rightarrow E \subseteq X \wedge$
$\forall_{(x,y) \in E \times X \setminus E}(x, y) \notin IND(B) \wedge \forall_{(x,y) \in E \times X^c}(x, y) \notin IND(B) \Leftrightarrow E \subseteq X \wedge$
$\forall_{(x,y) \in E \times U \setminus E}(x, y) \notin IND(B) \Leftrightarrow E \subseteq X \wedge E \in \mathcal{E} \Leftrightarrow E \in L$ by (c) and (d).
2–3. These can be shown analogously to the first point.

The representatives sets of rough regions can also be constructed using the families of equivalence classes of the concept and its complement.

Let $\widehat{E}_B \oplus \widehat{E'}_B = (\mathbf{x}_B, c + c') = (\mathbf{y}_B, c + c')$ for $\widehat{E}_B = (\mathbf{x}_B, c)$ and $\widehat{E'}_B = (\mathbf{y}_B, c')$ such that $\widehat{E}_B \simeq \widehat{E'}_B$.

Proposition 2. *For any $X \subseteq U$ in $AS_B = (U, IND(B))$ the following hold.*

1. $\widehat{POS}(AS_B, X) = \{\widehat{E}_B \in \mathcal{E}_X : \forall_{E' \in \mathcal{E}_{X^c}} \widehat{E}_B \not\simeq \widehat{E'}_B\}$;
2. $\widehat{BND}(AS_B, X) = \{\widehat{E}_B \oplus \widehat{E'}_B : E \in \mathcal{E}_X, E' \in \mathcal{E}_{X^c}, \widehat{E}_B \simeq \widehat{E'}_B\}$;
3. $\widehat{NEG}(AS_B, X) = \{\widehat{E}_B \in \mathcal{E}_{X^c} : \forall_{E' \in \mathcal{E}_X} \widehat{E}_B \not\simeq \widehat{E'}_B\}$.

Proof. 1. We obtain $\{\widehat{E}_B \in \mathcal{E}_X : \forall_{E' \in \mathcal{E}_{X^c}} \widehat{E}_B \not\simeq \widehat{E'}_B\} = \{\widehat{E}_B : E \subseteq \bigcup\{E \in \mathcal{E}_X : \forall_{E' \in \mathcal{E}_{X^c}} \widehat{E}_B \not\simeq \widehat{E'}_B\}\} = \{\widehat{E}_B : E \subseteq POS(AS_B, X)\} = \widehat{POS}(AS_B, X)$ by Definition 5 and Proposition 1.

2. **Lemma 1.** *For any $E \in \mathcal{E}_X$ and $E' \in \mathcal{E}_{X^c}$ the following holds $\widehat{E}_B \oplus \widehat{E'}_B = \widehat{(E \cup E')}_B$.*
   The lemma can easily be proven by the definition of the $\oplus$ operation.
   We obtain $\{\widehat{E}_B \oplus \widehat{E'}_B : E \in \mathcal{E}_X, E' \in \mathcal{E}_{X^c}, \widehat{E}_B \simeq \widehat{E'}_B\} = \{\widehat{E}_B \oplus \widehat{E'}_B : E \cup E' \subseteq \bigcup\{E \cup E' : E \in \mathcal{E}_X, E' \in \mathcal{E}_{X^c} : \widehat{E'}_B \simeq \widehat{E'}_B\} = \{\widehat{(E \cup E')}_B : E \cup E' \subseteq BND(AS_B, X)\} = \widehat{BND}(AS_B, X))$ by the lemma, Definition 5 and Proposition 1.
3. This can be shown analogously to the first point.

2.3 Classification Rules Generated from Representatives Sets

In rough set theory classification rules are usually generated based on one of the lower or upper approximations.

Definition 6. *(rough approximation rule) A rough approximation rule in $AS_B = (U, IND(B))$ is an expression of the form $\alpha \to \beta$, where $\alpha = \bigwedge_{i=1}^{k}(a_i, v_i), \beta = (app, v), a_i \in A$ ($v_i \in V_{a_i}$) are condition attributes and app ($app \in \{lower, upper\}$, $V_{app} = \{0, 1\}$) is the decision attribute such that $d(x) = 1 \Leftrightarrow x \in APP(AS_B, X)$ for any $x \in U$ and a given $X \subseteq U$.*

Let $match_X(r)$ be the matching of a rule r over $X \subseteq U$, i.e. the set of objects from X that satisfy the rule premise. Let also $\widehat{X}_B = \{\widehat{E}_B : E \in \mathcal{E}_X\}$ be the representatives set of $X \subseteq U$.

Definition 7. *(matching of rule over representatives set) The matching of a rule r over a representatives set $\widehat{X}_B$, denoted by $match_{\widehat{X}_B}(r)$, is defined as follows $(\mathbf{x}_B, c) \in match_{\widehat{X}_B}(r) \Leftrightarrow x \in match_X(r)$.*

Proposition 3. *For any rough approximation rule r the following holds*

$$match_{\widehat{X}_B}(r) = \widehat{(match_X(r))}_B.$$

Proof. We obtain $(\mathbf{x}_B, c) \in \widehat{(match_X(r))} \Leftrightarrow (\mathbf{x}_B, c) \in \{\widehat{E}_B : E \in \mathcal{E}_{match_X(r)}\} \Leftrightarrow x \in \bigcup\{E : E \in \mathcal{E}_{match_X(r)}\} \Leftrightarrow x \in match_X(r) \Leftrightarrow (\mathbf{x}_B, c) \in match_{\widehat{X}_B}(r)$ by Definition 7.

Let $card_{\widehat{X}_B}(match_{\widehat{X}_B}(r)) = \sum_{(\mathbf{x}_B, c) \in match_{\widehat{X}_B}(r)} c$.

Proposition 4. *For any rough approximation rule r the following holds*

$$card_{\widehat{X}_B}(match_{\widehat{X}_B}(r)) = |match_X(r)|.$$

Proof. Let $\{x_1, \ldots, x_n\} \subseteq X$ be the set of objects from X matching a given rule r. We have $match_X(r) = \{x_1, \ldots, x_n\} = \{x_1^1, \ldots, x_{n_1}^1, x_1^2, \ldots, x_{n_2}^2, \ldots, x_1^k, \ldots, x_{n_k}^k\} = \bigcup_{i=1}^{k} E^i$, where $E^i = \{x_1^i, \ldots, x_{n_1}^i\}$ is an equivalence class. Hence, $|match_X(r)| = \sum_{i=1}^{k} |E^i|$. We also have $card_{\widehat{X}_B}(match_{\widehat{X}_B}(r)) = \sum_{(\mathbf{x}_B, c) \in match_{\widehat{X}_B}(r)} c = \sum_{E \in \mathcal{E}_{match_X(r)}} |E| = \sum_{i=1}^{k} |E^i|$.

The below example illustrates the notions introduced in this section.

Example 1. Given a data table of patients who are suspected to be sick with flu.

$U \setminus A$	temperature	headache	weakness	nausea	flu
1	very high	yes	yes	no	yes
2	normal	no	no	no	no
3	high	no	no	no	no
4	normal	no	yes	no	yes
5	normal	no	yes	no	no
6	high	yes	no	yes	yes
7	very high	no	no	no	no
8	normal	yes	yes	yes	yes

Consider the approximation space $AS_B = (U, IND(B))$, where $B = \{headache, weakness\}$, and the concept of patients sick with flu represented by the set $X = \{1, 4, 6, 8\}$.
We obtain $U/IND(B) = \{\{1, 8\}, \{2, 3, 7\}, \{4, 5\}, \{6\}\}, POS(AS_B, X) = \{1, 6, 8\}$, $BND(AS_B, X) = \{4, 5\}$ and $NEG(AS_B, X) = \{2, 3, 7\}$.
Let $\mathbf{x}_B^1 = (yes, yes), \mathbf{x}_B^2 = (no, yes), \mathbf{x}_B^3 = (yes, no), \mathbf{y}_B^1 = (no, no), \mathbf{y}_B^2 = (no, yes)$. We obtain $\widehat{X}_B = \{(\mathbf{x}_B^1, 2), (\mathbf{x}_B^2, 1), (\mathbf{x}_B^3, 1)\}, \widehat{X^c}_B = \{(\mathbf{y}_B^1, 3), (\mathbf{y}_B^2, 1)\}$, $\widehat{POS}(AS_B, X) = \{(\mathbf{x}_B^1, 2), (\mathbf{x}_B^3, 1)\}, \widehat{BND}(AS_B, X) = \{(\mathbf{x}_B^2, 1) \oplus (\mathbf{y}_B^2, 1)\}$ and $\widehat{NEG}(AS_B, X)) = \{(\mathbf{y}_B^2, 1)\}$.
Consider the rule $r : (weakness, yes) \rightarrow (lower, 1)$. Let $P = POS(AS_B, X)$ and $N = BND(AS_B, X) \cup NEG(AS_B, X)$. We obtain $|match_P(r)| = |\{1, 8\}| = 2$ and $|match_N(r)| = |\{4, 5\}| = 2$.
We obtain $card_{\widehat{P}_B}(match_{\widehat{P}_B}(r)) = card_{\widehat{P}_B}(\{(\mathbf{x}_B^1, 2)\}) = 2$, $card_{\widehat{N}_B}(match_{\widehat{N}_B}(r))$ $= card_{\widehat{N}_B}(\{(\mathbf{x}'_B, 2)\}) = 2$, where $(\mathbf{x}'_B, 2) = (\mathbf{x}_B^2, 1) \oplus (\mathbf{y}_B^2, 1)$.

3 Algorithms for Computing Representatives of Rough Regions and Generating Classification Rules

This section proposes a sort-based algorithm for computing representatives sets of rough regions. It also adapts a sequential covering algorithm to generate classification rules from representatives sets.

Firstly, a sort-based algorithm for computing lower and upper approximations will be recalled [4]. It can be outlined as follows.

1. Given an information system $IS = (U, A)$, a subset $B \subseteq A$ and a concept $X \subseteq U$.
2. Sort objects from U (represented by B-information vectors) in the lexicographical order.
3. Form the equivalence classes based on objects that are equal according to the order.

4. Form approximations: If an equivalence class is included in X, then add the class to the lower approximation; If an equivalence class and X have at least one common element, then add the class to the upper approximation.

The algorithm needs $O(|B||U|log|U|)$ time to sort objects, and $O(|U|)$ time to form approximations, assuming that the information on the membership to the concept is associated with each object of the universe.

The algorithm called *ComputeRegionRep* for computing representatives sets of rough regions is proposed.

Let $(\mathbf{x}_B, c) \preceq (\mathbf{y}_B, c') \Leftrightarrow \forall_{a \in B} a(x) \leq a(y)$ and $(\mathbf{x}_B, c) \prec (\mathbf{y}_B, c') \Leftrightarrow (\mathbf{x}_B, c) \preceq (\mathbf{y}_B, c') \wedge \exists_{a \in B} a(x) < a(y)$.

Algorithm 1: *ComputeRegionRep*

Data: $AS_B = (X \cup X^c, IND(B))$ – an approximation space with the specified concept X to be approximated;
Result: $(repPOS, repBND, repNEG)$ – a triple of the representatives sets of the positive, boundary, and negative regions;

```
begin
    repPOS := ∅; repBND := ∅; repNEG := ∅;
    Y := mergeSortUnique(X, B); Y' := mergeSortUnique(X^c, B);
    i := 1; j := 1;
    while i < |Y| or j < |Y'| do
        if j > |Y'| then
            repPOS := repPOS ∪ {Y[i], ..., Y[|Y|]}; i = |Y|;
        end
        else if i > |Y| then
            repNEG := repNEG ∪ {Y'[j], ..., Y'[|Y'|]}; j = |Y'|;
        end
        else if Y[i] ≺ Y'[j] then
            repPOS := repPOS ∪ {Y[i]}; i := i + 1;
        end
        else if Y[i] ≃ Y'[j] then
            repBND := repBND ∪ {Y[i] ⊕ Y'[j]}; i := i + 1; j := j + 1;
        end
        else
            repNEG := repNEG ∪ {Y'[j]}; j := j + 1;
        end
    end
end
```

Algorithm 1 needs not more than $O(|B||X|log|X|) + O(|B||X^c|log|X^c|)$ time to compute representatives of equivalence classes for sets X and X^c. The equality holds when each object is indiscernible with itself only. The cost of joining the representatives is $O(|U|)$. We have $O(|B||X|log|X|) + O(|B||X^c|log|X^c|) + O(|U|) \geq O(|B||U|log|U|)$. The equality holds for $|X| = |X^c|$.

The $mergeSortUnique(X)$ function is defined as follows.
1. Form the initial representative of X, i.e. $\widehat{X}_B = \{(\mathbf{x}_B, 1) : x \in X\}$.
2. Order $\widehat{X}_B$ using the merge sort algorithm that applies the following comparison function

$$compare((\mathbf{x}_B, c), (\mathbf{x}'_B, c')) = \begin{cases} (\mathbf{x}_B, c), & (\mathbf{x}_B, c) \prec (\mathbf{x}'_B, c'); \\ (\mathbf{x}'_B, c'), & (\mathbf{x}'_B, c') \prec (\mathbf{x}_B, c); \\ (\mathbf{x}_B, c) \oplus (\mathbf{x}'_B, c'), & (\mathbf{x}_B, c) \simeq (\mathbf{x}'_B, c'). \end{cases}$$

The algorithm called *GenerateRuleSet* adapts sequential covering one to generate classification rules from representatives sets.

Algorithm 2: *GenerateRuleSet*

Data: $repPOS, repBND, repNEG$ – the representatives sets of the three regions; app – the type of approximation;
Result: RS – a rule set generated over APP;
begin
  $RS := \emptyset$; $\beta := (app, 1)$;
  $P := repPOS$; $N := repNEG$;
  if $app = lower$ **then** $N := N \cup repBND$;;
  else $P := P \cup repBND$;
  ;
  while $P \neq \emptyset$ **do**
    $\alpha := \emptyset$; $N' := N$;
    while $N' \neq \emptyset$ **do**
      $c := findBestCandidate(P, N')$;
      $\alpha := \alpha \wedge c$; $N' := match_{N'}(\alpha \rightarrow \beta)$;
    end
    $P := P \setminus match_P(\alpha \rightarrow \beta)$;
  end
  $RS := RS \cup \{\alpha \rightarrow \beta\}$;
end

The $findBestCandidate$ function uses any measure that calculates the rule quality based on its matching.

4 Experiments

This section describes experimental research that concerns computation of rough regions and their representatives sets, as well as generation of classification rules based on them.

Six datasets (Table 1) taken from the UCI Repository (archive.ics.uci.edu/ml) were used in the experimental research. The approach was implemented in C++ and tested using a laptop with Intel Core i5, 2.3 GHz, 4 GB RAM, Windows 7.

To compute rough regions a slight modification of the algorithm described in the beginning of Section 3 was used. Equivalence classes are generated in the

Function 3: $findBestCandidate$

Data: P, N – the representatives sets of the approximation and its complement; $\alpha \to \beta$ – the current rule; $q\_measure$ – quality measure;
Result: c – the best candidate to add to the rule;
begin
  $p := card_P(match_P(\alpha \to \beta))$; $n := card_N(match_N(\alpha \to \beta))$;
  $q := q\_measure(p, q)$; $C := generateCandidates(\alpha \to \beta)$;
  foreach $c' \in C$ **do**
    $p' := card_P(match_P(\alpha \wedge c' \to \beta))$; $n' := card_N(match_N(\alpha \wedge c' \to \beta))$;
    $q' := q\_measure(p', q')$;
    if $q' > q$ **then**
      $c := c'$; $q := q'$;
    end
  end
end

Table 1. Characteristics of datasets.

symbol	dataset	no. attributes	no. objects
D1	Bank Marketing	17	45K
D2	Connect-4	43	67K
D3	Skin Segmentation	4	245K
D4	KDD Cup 1999 Data (10%)	42	494K
D5	Record Linkage Comparison Patterns (10%)	10	574K
D6	Poker Hand (testing data)	11	1000K

same way, but instead of approximations, regions are computed. Before generation of regions, each object is assigned the label informing whether the object belongs to the concept. Thanks to this, it is enough to scan the universe once to check what region a given equivalence class belongs to. Representatives set of rough regions were generated using Algorithm 1.

Table 2 shows the run-time (given in seconds) for generating regions (standard approach, denoted by std.) and their representatives (representative-based approach, denoted by rep.). For both the approaches the same distribution of data was used (the concept and its complement are equinumerous, denoted by 1-1). Since for the proposed approach a change in data distribution may influences the run-time (see the time complexity analysis in Section 3), additional two distributions of data were used (in the first (second) distribution the concept is three times smaller (bigger) than its complement, denoted by 1-3 (3-1)). The table includes also the size of each representatives set given in percentage (the ratio between the number of representatives and the cardinality of the universe multiplied by 100). Independent of the data distribution the regions and their representatives do not change. That is why the size of representatives sets is given once. For each dataset four tests were carried out: the subset of the

attribute set included 25%, 50%, 75%, 100% of its attributes (Attributes were taken in the order they appear in the datasets).

Table 2. Computation of regions and their representatives sets

		25	50	75	100		25	50	75	100
std. 1-1	*D1*	0.39	0.41	0.41	0.42	*D2*	0.73	0.89	0.95	0.95
rep. 1-1		**0.34**[1]	0.53	0.53	0.53		**0.48**	**0.72**	1.42	1.63
rep. 1-3		0.35	0.54	0.54	0.54		0.48	0.72	1.48	1.76
rep. 3-1		0.35	0.53	0.53	0.55		0.47	0.73	1.47	1.72
rep.(size)		8.04	90.79	100	100		0.98	19.35	78.62	100
std. 1-1	*D3*	1.50	1.59	1.65	1.67	*D4*	5.20	6.13	8.60	9.92
rep. 1-1		**1.27**	**1.39**	**1.47**	**1.48**		**3.09**	**3.35**	**6.60**	**7.85**
rep. 1-3		1.27	1.41	1.50	1.52		3.19	3.41	6.78	8.12
rep. 3-1		1.26	1.411	1.50	1.51		3.18	3.47	6.72	8.02
rep.(size)		0.10	8.50	20.99	20.99		11.58	11.62	20.30	29.47
std. 1-1	*D5*	4.24	5.40	6.08	6.80	*D6*	7.75	9.08	9.89	9.97
rep. 1-1		**2.96**	**3.99**	**4.52**	**4.70**		**5.84**	**8.91**	13.14	13.18
rep. 1-3		3.0	4.06	4.51	4.83		5.92	9.35	13.67	13.84
rep. 3-1		2.98	4.07	4.564	4.88		5.93	9.26	13.64	13.80
rep.(size)		0.04	0.57	01.16	01.54		<0.01	01.06	92.72	99.79

The results show that if a representatives set is decidedly smaller than the universe, then the run-time is shorter. When the size is close to that of the universe, especially when they are equal, the run-time may be longer. That reason is that Algorithm 1 needs additional operations compared with the standard approach, namely it joins representatives if they are constructed based on the same information vector.

A small size of a representatives set does not always imply a considerably shorter run-time (see e.g. $D1$-25). According to sort-based algorithm we need $log|U|$ steps to sort $|U|$ objects. In the proposed approach, the number of representatives to sort is fewer than or equal to $|U|$. The run-time may depend on how fast (i.e. in which steps) representatives are joined. For example, if most representatives are joined in the last step, then the run-time is inconsiderably shorter.

The run-times obtained for data distributions 1-3 and 3-1 are slightly longer than those for 1-1. Namely, on average we have respectively 3.25% and 2.73% longer run-times. In theory, based on the analysis of the time complexity, we should obtain the same results for data distributions 1-3 and 3-1 (about 3.34% longer run-time than for 1-1[2]). In practice, the run-time depends, in a sense, on

[1] A run-time obtained by the proposed approach that is shorter than that for the standard one is written in bold.

[2] The value is computed according to the formula $(a|U|log(a|U|) + (1-a)|U|log((1-a)|U|))/(2 * 0.5|U|log(0.5|U|))$, where $a \in (0,1)$ defines the data distribution.

data distribution since equivalence classes are computed separately for the concept and its complement. Therefore, a change in the concept may cause a change in its equivalence classes. Thus, the run-time of computing representatives sets for different concepts may vary.

To generate classification rules from the whole regions, sequential covering algorithm was used. The general mechanism of rule generation is the same as in Algorithm 2 that is used for generating rules from representatives sets. In both cases, the coverage measure was used to evaluate the quality of rules. Rules were generated from the upper approximation of a concept (the union of positive and boundary regions) since it is always different form an empty set.

Table 3 presents results of rule generation for data distribution 1-1 as well as the size of rule sets, denoted by $|RS|$. Since the algorithms for rule generation are not so efficient as those for region computation, then some experiments that were time-consuming (i.e., experiments needing over 30 minutes to finish computations – denoted by *) were interrupted.

Table 3. Ruleset generation

		25	50	75	100		25	50	75	100
std. 1-1	*D1*	13.88	*	*	*	*D2*	1.42	49.62	169.07	347.03
rep. 1-1		**3.09**	*	*	*		**0.07**	**7.52**	**123.83**	**344.12**
$\|RS\|$		1180	?	?	?		85	157	288	419
std. 1-1	*D3*	61.78	*	*	*	*D4*	*	*	*	*
rep. 1-1		**0.16**	**1321.86**	*	*		*	*	*	*
$\|RS\|$		256	13323	?	?		?	?	?	?
std. 1-1	*D5*	5.67	162.63	285.76	430.72	*D6*	3.81	5.74	*	*
rep. 1-1		**0.02**	**9.27**	**26.55**	**52.87**		**<0.001**	**0.46**	*	*
$\|RS\|$		150	1663	2206	2671		4	4	?	?

The results clearly show that rule generation from representative sets is decidedly less time-consuming. On average, the proposed approach is 2.7 times faster. If the size of a representatives set is close to that of the universe (see $D2$-75 and $D2$-100), the run-time is also closer to that obtained for the whole universe. However, unlike for region computation, the algorithm for generating rules from representatives sets does not perform additional operations. Thanks to this, the time needed for computing rules from a representatives set of the size equal to that of the universe should not be longer than in the standard case (see $D2$-100).

A relatively small representatives set does not guarantee a short run-time. For example, it is needed over 22 minutes to accomplish computations for $D3$-50. The reason is a relatively big number of rules (13323). Namely, one rule is generated for each object of the set to be described by rules.

5 Conclusion

This paper has proposed an alternative approach to computation of rough regions for generating classification rules. Representative sets that replace positive, boundary, and negative regions include essential information needed during rule generation.

The approach enables to speed up computations of rough regions for data divisible into a relatively low number of equivalence classes. Compared with the standard approach the run-time of rule generation is significantly shorter if a representatives set is small. For representatives sets close to the whole universe in terms of size results are slightly better.

Thanks to employing an efficient sort algorithm for computing representatives set of rough regions the approach can find application in processing large datasets, especially those considered in the field of Big Data. An ease of adaptation of sequential covering algorithm to generating rules from representatives of rough regions shows that the approach can be combined with any rule generation algorithm that uses the matching measure.

References

1. Bonikowski, Z.: Algebraic structures of rough sets in representative approximation spaces. Electr. Notes Theor. Comput. Sci. 82(4), 52–63 (2003)
2. Demri, S., Orłowska, E.: Logical analysis of indiscernibility. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis, Stud. Fuzziness Soft Comput., vol. 13, pp. 347–380. Physica Verlag (1998)
3. Geng, L., Chan, C.W.: An algorithm for automatic generation of a case base from a database using similarity-based rough approximation. In: Abraham, A., Koppen, M. (eds.) HIS. pp. 571–582. Advances in Soft Computing, Physica-Verlag (2001)
4. Nguyen, S.H., Nguyen, H.S.: Some efficient algorithms for rough set methods. In: IPMU'96. vol. 3, pp. 1451–1456. Physica-Verlag (2001)
5. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic, Dordrecht (1991)
6. Stepaniuk, J.: Rough-Granular Computing in Knowledge Discovery and Data Mining. Studies in Computational Intelligence 152, Springer-Verlag, Berlin-Heidelberg (2008)
7. Zhang, B., Min, F., Ciucci, D.: Representative-based classification through covering-based neighborhood rough sets. Appl. Intell. 43(4), 840–854 (2015)