CAHIER DU LAMSADE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
(Université Paris-Dauphine)
Unité de Recherche Associée au CNRS n° 825

DISCOVERY-ORIENTED INDUCTION OF DECISION RULES

Robert MIEŃKO [1]
Jerzy STEFANOWSKI [1]
Khaled TOUMI [2]
Daniel VANDERPOOTEN [2]

CAHIER N° 141
septembre 1996

received: May 1996.

[1] *Politechnika Poznańska, Instytut Informatyki, ul. Piotrowo 3a, 60-965 Poznań, Poland (e-mail: {mienko,stefanj}@pozn1v.put.poznan.pl).*

[2] *Université Paris-Dauphine, LAMSADE, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France (e-mail: {toumi,vdp}@lamsade.dauphine.fr).*

Table of Contents

Induction de règles de décision pour l'extraction de connaissances

Résumé

Deux perspectives principales pour l'induction de règles de décision à partir d'exemples peuvent être envisagées. Outre la perspective classique de « l'induction orientée classification », dont l'objet est de bâtir un classifieur, nous distinguons et discutons « l'induction orientée extraction de connaissances » dont l'objectif est d'extraire des règles « intéressantes » et utiles pour l'utilisateur. La signification d'« intéressant » doit être définie selon le niveau d'expertise de l'utilisateur et/ou ses exigences. Compte-tenu de la diversité des objectifs respectifs, il nous semble impératif de concevoir différemment les approches pour l'induction orientée extraction de connaissances et celles, plus classiques, pour l'induction orientée classification. Une approche spécifique visant à l'extraction de connaissances est présentée dans ce papier. Une implémentation informatique nous a permis d'illustrer l'intérêt de cette approche dans le cadre de plusieurs expérimentations.

Mots clés: Apprentissage par induction, extraction de connaissances, règles de décision, classification.

Discovery-oriented induction of decision rules

Abstract

Two main perspectives for inducing decision rules from examples can be envisaged. Besides the classical perspective of 'classification-oriented induction' whose objective is to build a classifier, we distinguish and discuss 'discovery-oriented induction' whose purpose is to extract rules 'interesting' and useful to different kinds of users. The meaning of 'interesting' clearly depends on the user's level of expertise and requirements. According to the diversity of the respective objectives, we claim that approaches for discovery-oriented induction must be conceived quite differently from well-known approaches for classification-oriented induction. Hence, a specific approach for discovery-oriented induction is proposed. The results of several experiments with an implementation of our approach illustrate its usefulness for discovery purposes.

Keywords: Inductive learning, knowledge discovery, decision rules, data mining, classification.

1 Introduction

This paper deals with induction of *decision rules* in data sets representing experience in certain domains. It is assumed that data sets contain information about a set of *objects* described by a set of *attributes*. The problem consists in finding rules that determine whether an object belongs to a particular subset called a *decision class* or *concept*. It is assumed that the definition of this class is known (e.g., given by experts or users). Hence, this is a case of the so-called *supervised learning*. The objects analysed are called *learning examples*. The rules considered are logical expressions of the following form:

$$IF <conditions> \ THEN <decision \ class>$$

where conditions are formed as a conjunction of elementary tests on values of attributes.

Induction of decision rules can be performed because of different aims. The most common ones are connected with:

- classification-oriented induction,
- discovery-oriented induction.

The aim of *classification-oriented induction* is to find automatically, from the set of learning examples, a collection of decision rules which will be used to *classify* future examples. This problem has been, in fact, extensively studied in *Machine Learning* literature and several approaches for deriving such *classification rules* have been proposed [17]. Some of them provide a set of rules learned directly from examples (see, e.g., the well-known algorithms *AQ* [9, 11, 10], *CN2* [2], *PVM* [21], ...), while other approaches first generate decision trees and then transform the branches of the trees into rules (as done in Quinlan's system *C4.5* [14]).

The main criterion for evaluating the quality of a set of classification rules is the *classification accuracy rate* (or conversely the *misclassification error rate* - see, e.g., [22]). Several comparative studies, based on experiments using this criterion, have shown that most of these algorithms give similar rather good results. It must be noticed that other classification systems using, e.g., neural networks or statistical approaches perform at least as well, and sometimes better than rule-based systems [22, 17]. However, it is often claimed in favour of the latter systems that they provide a *symbolic representation* of *classification knowledge* which is useful for justification of classification results (see [9, 10]).

The aim of *discovery-oriented induction* is to extract, from data sets, information patterns and regularities *interesting* and *useful* to different kinds of users (data analysts, experts, ...). Discovered patterns and regularities, if they are represented in the form of rules, will be further referred to as *'interesting' rules*. Such 'interesting' rules

can help in understanding and explaining relationships between values of attributes and definitions of decision classes. Clearly, the meaning of 'interesting' must be modulated according to the user's expectations and requirements. In general, however, interesting rules should correspond to strong and simple patterns.

Discovery-oriented induction is not so well studied as classification-oriented induction, although it has been raising an increasing attention in the last years due to the development of the *Knowledge Discovery* research field [4, 10, 1, 23].

As the well-known rule induction algorithms have been introduced taking into account classification aims, the results of their direct application to discover data patterns may be quite unsatisfactory. Moreover, nearly all of these algorithms give as a result a *minimum set* of rules. It must be stressed that this set represents only a *limited part* of the *most 'interesting'* decision rules. Other 'interesting' rules may still remain hidden in the data set. Conversely, a minimum set of rules may include very specific rules with no real interest. Therefore, we believe it is necessary to develop specific approaches for discovery-oriented induction.

The aim of this paper is to underline some specificities of discovery-oriented induction and to propose an approach for extracting 'interesting' rules from a set of examples.

The paper is organized as follows. In section 2, motivations for discovering other types of rules than for classification aims, are discussed more precisely. After introducing some basic concepts in section 3, the criteria for evaluating 'interesting rules' are presented in section 4. In section 5, the algorithm for getting the set of 'interesting' rules is presented. Some computational experiments are described in section 6. Final remarks are given in the last section.

2 Specificities of discovery-oriented induction

The aim of discovery-oriented induction is to extract 'interesting' rules, i.e. rules which are of interest and use for different kinds of users (decision analysts, experts, ...). Actually, discovery-oriented induction is much more difficult to define than classification-oriented induction. Indeed, the meaning of terms like 'interesting rules' or 'potentially useful information patterns' [4] is not so obvious and depends on the interests and expertise level of users. An advanced user or an expert will certainly look for other patterns than a novice and should be able to direct and constrain somehow the discovery process.

In spite of these difficulties, in discovery-oriented induction, one is usually interested in getting a set of decision rules which are (see, e.g., [4, 13, 23]):

a) strong, i.e. refer to a large number of learning examples,

b) **simple**, i.e. whose condition parts consist of a rather limited number of elementary conditions easy to interpret by the user,

c) **consistent**, i.e. the relationship between its condition part and decision part is sufficiently well supported.

Let us emphasize that in discovery-oriented induction, each rule is evaluated individually and independently as a possible representant of a specific pattern. In classification-oriented induction, rules are parts of a system; hence, the evaluation refers here to the complete set of rules.

One can notice that it is not so easy to define criteria representing *requirements* **a**, **b**, **c** and above all to consider all of them together. In case of induction algorithms creating classification rules, there is in fact one dominant criterion expressing misclassification error rate. For discovery systems none of the criteria is dominant and they must be treated in a different way depending on the current context.

A crucial issue refers to the way of getting such 'interesting' rules. The well-known approaches for inducing classification rules have been constructed taking into account only classification point of view. Most of them are focused on inducing a *minimum* (smallest due to some additional criteria [9, 17]) set of rules using a *greedy heuristic strategy*. This strategy consists in creating a first rule by choosing sequentially the 'best' elementary conditions according to some heuristic criteria. Then, learning examples that match this rule are removed from consideration. While there are still some significant undescribed examples, the procedure is repeated.

The decision rules obtained in this way, although work well in classification, may not be easy to interpret by the user. Such difficulties in interpreting and understanding classification rules are particularly clear in the case of *decision lists* (as these created by the *C4.5* or *CN2* systems) where an interpretation of each rule depends on its position in the list.

Another disadvantage, even more important from a discovery point of view, is connected with the fact that a minimum set of rules contains only a *limited part* of 'interesting' rules. Conversely, such a set may include specific rules of no interest. The perspective is quite different in discovery-oriented induction where *all* 'interesting' and *only* 'interesting' rules should be extracted.

The limitations of existing algorithms discussed above lead us to propose an alternative approach which aims at inducing only 'interesting' rules and all of them by focusing on the rules that satisfy requirements, like **a**, **b**, **c**. In this approach, we use an algorithm originally introduced by two of the authors in [20]. The algorithm progressively generates rules of increasing size. The exploration of the rule space is controlled by incorporating one or several *stopping conditions* (connected with *user's requirements*). The stopping conditions guarantee desirable properties of the rules and significantly reduce the computational costs of the algorithm.

Consulting the literature one can notice that similar motivations were also the starting points for two other approaches to non-standard rule discovery developed independently in last years by [15, 16] (*Brute* system) and [8]. However, our approach uses a different strategy for exploring the space of rules. Moreover, as it will be shown further, our algorithm can be easily adapted to the users' requirements.

3 Basic concepts

It is assumed that classification rules are discovered from examples represented in an attribute-value form. Let X be a set of objects or examples partitioned into m classes $X_1, \ldots, X_m$. Each class X_i $(i = 1, \ldots, m)$ is considered independently so as to be described on the basis of its *positive examples* (objects from X_i) and *negative examples* (objects from $X \setminus X_i$). In the following, X_i will represent the *decision concept* K to be described.
Let us consider the following concepts and notations:

- A *selector* is a basic statement representing an elementary condition which can be checked for any $x \in X$. In most systems learning from *attribute-value* representations, selectors are expressed in the form $<$*attribute rel set-of-values*$>$ where *rel* stands, e.g., for $=, \neq, \leq, \ldots$ or $\in$ and *set-of-values* is a specific value or a subset of values. A selector s can be interpreted as a mapping $s : X \rightarrow \{\text{true, false}\}$.

- A *complex* C, of size q, is a conjunction of q selectors: $C = s_1 \wedge s_2 \wedge \cdots \wedge s_q$. The size of complex C will be denoted by *Size*(C).

- The *cover* of a complex C, denoted by $[C]$, is the subset of examples which satisfy the conditions represented by C. Formally, we have: $[C] = \{x \in X : C(x) = true\}$.

- Considering the concept K to be described, $[C]_K^+ = [C] \cap K$ denotes the set of *positive examples* covered by C and $[C]_K^- = [C] \cap (X \setminus K)$ denotes the set of *negative examples* covered by C.

- A *rule* r (which partially describes K) is an assertion of the form

$$\textit{if } R(x) \textit{ then } (x \in K)$$

  where R is a complex $s_1 \wedge s_2 \wedge \cdots \wedge s_q$, satisfying $[R]_K^+ \neq \emptyset$.
  A rule r is thus characterized by its condition part R and the concept K described by r.

- A rule r is *discriminant* (i.e. distinguishes positive examples belonging to K from negative ones) if its condition part $R = s_1 \wedge s_2 \wedge \cdots \wedge s_q$ is:
  - *consistent*: $[R]_K^- = \emptyset$,
  - *minimal*: removing any selector s_j from R would result in a complex which is no longer consistent.

In some data sets (in particular if they contain inconsistent examples), discovery systems could find only few discriminant rules which are 'interesting' (in the sense presented in section 2). In such cases, it may make sense to look for *partly discriminant* rules. These are rules which besides positive examples could cover a limited number of negative ones. They are characterized by a coefficient called *level of discrimination* defined as:

$$D(R) = \frac{|\,[R]_K^+\,|}{|\,[R]\,|}$$

where $|\,.\,|$ denotes the cardinality of a set. The above definition for discriminant rules is thus generalized as follows:

- a rule r is *partly discriminant*, considering a threshold d $(0 \leq d \leq 1)$, if its condition part $R = s_1 \wedge s_2 \wedge \cdots \wedge s_q$ is:
  - *partly consistent*: $D(R) \geq d$,
  - *minimal*: removing any selector s_j from R would result in a complex which is no longer partly consistent.

When considering (strictly) discriminant rules, we just set $d = 1$.

4 Evaluation criteria for 'interesting' rules

In order to evaluate discovered rules several measures could be used (see, e.g., [9, 8, 21]). In this paper, the rules are characterized by measures connected with requirements **a**, **b**, **c** (described in section 2). For a given rule r with condition part R, the following measures are considered:

a) The *strength* of r, denoted by *Strength*(r), which can be expressed in one of the following ways:

- *absolute strength*, in which case we have:

$$Strength(r) = |\,[R]_K^+\,|$$

– *relative strength*, in which case we have:

$$Strength(r) = \frac{\mid [R]_K^+ \mid}{\mid K \mid}$$

b) the *length* of the rule r, denoted by $Length(r)$:

$$Length(r) = Size(R)$$

c) the *level of discrimination* of r, denoted by $Disc(r)$:

$$Disc(r) = D(R) = \frac{\mid [R]_K^+ \mid}{\mid [R] \mid}$$

When considering 'interesting' rules as a set of rules, we must also take into account the two following requirements:

d) the number of rules must be acceptable: this number should indeed not be too high owing to the limited cognitive abilities of the user (this largely depends on the user's level of expertise),

e) the classification accuracy rate must be acceptable (compared with classification accuracy rates obtained by classifiers).

As to requirement **e**, we insist again on the fact that classification accuracy is not the most important aspect for discovery-oriented induction. However, rules which are totally unable to describe the set of learning examples would be doubtful.

5 An algorithm for extracting 'interesting' rules

The basic purpose of the approach considered in this paper is to extract, from the given learning examples, the set of all 'interesting' rules which satisfy the user's defined requirements. This is achieved by *exploring* the rule space imposing restrictions related to *requirements* **a**, **b** and **c**. This involves the *definition of thresholds* respectively for the strength, length and level of discrimination of the rules to be generated. The resulting rules are then examined considering requirements **d** and **e**. Notice that the thresholds should be defined considering the user's requirements. However, such parameters may seem rather technical and depend on each data set. We shall show in section 6 how to set these thresholds by testing different values.

The exploration of the rule space is performed using an algorithm which is *repeated iteratively* for each concept K to be described. The main part of the algorithm,

originally introduced in [20], is based on a *breadth-first* strategy which generates rules of increasing size starting from the shortest ones.

The strategy begins with the initial rule having an empty condition part. During the search process this empty complex is extended with selectors from the list of allowed selectors. The extended complexes are evaluated as candidates for being condition parts of rules.

Considering a complex C with covers positive and negative examples ($[C]_K^+ \neq \emptyset$ and $[C]_K^- \neq \emptyset$), the determination of the cover of a new complex C' derived from C by including a new selector is quite straightforward. Indeed if $C' = C \wedge s_l$, we have :

$$[C']_K^+ = [C]_K^+ \cap [s_l]_K^+ \text{ and } [C']_K^- = [C]_K^- \cap [s_l]_K^-$$

which makes the evaluation of a candidate complex, derived from another complex by including a new selector, particularly easy.

The above feature is also particularly important when we consider applying discovery techniques to *large, real-world databases*, as done in the *data mining* field (see [1, 8, 4]). Let us notice that to evaluate all extensions of a complex C, we only need to look at the cover of C and not at the entire data set. This 'zooming' property of the considered search strategy can reduce the response time connected with direct access to data. Direct data access operations are only needed in the initial stage of discovery in order to create covers of chosen selectors. Then, the results are kept in the main memory and progressively re-used in the following steps of the search strategy.

Hence, the main breadth-first search strategy phase follows the initial phase where the list of available and allowed (according to the user's requirements) selectors is created. Then, the sets of objects covered by each selector are identified, i.e. for each selector we determine $[s]_K^+$ and $[s]_K^-$.

The creation of the selectors may be performed in a more or less sophisticated way depending on the type of attributes under consideration, i.e. whether they are nominal or ordered ones (see, e.g., [13, 1, 8] or [21, 14]) [1].

The main part of the algorithm, i.e. phase of breadth-first search is presented in pseudo-code:

[1]The typical forms of selectors *attr rel val* have been discussed in Section 3. Let us add that creation and evaluation of selectors are easier when one uses information about histograms of value occurrence for given attributes [1, 8, 4]

```
Procedure Explore(d: discriminant_threshold; SC: stopping_conditions; var R: set_of_rules)
begin
   R ← ∅
   for each available selector s do
   begin
      if [s]_K^+ = ∅ or s satisfies SC then discard s;
      if D(s) ≥ d then R ← R ∪ {s} and discard s
   end;
   form a queue with all the remaining selectors s_1, ..., s_n;

   while the queue is not empty do
   begin
      remove the first complex C from the queue;
      let h be the highest index of the selectors involved in C;
      generate all the complexes C ∧ s_{h+1}, C ∧ s_{h+2}, ..., C ∧ s_n;
      let C be the set of these complexes;
      for each C' ∈ C do
      begin
         if [C']_K^+ = ∅ or C' satisfies SC then C ← C \ {C'};
         if D(C') ≥ d then
         begin
            if C' is minimal then R ← R ∪ {C'};
            C ← C \ {C'}
         end;
      end;
      place all the complexes from C at the end of the queue
   end
end
```

The above algorithm is very general and can be improved in many ways to reduce the exploration. These improvements range from general heuristic ideas (e.g., ordering the selectors in decreasing order considering the number of objects covered) to small implementation details (e.g., it is unnecessary to put complexes of the form $C' = C \wedge s_n$ in the queue since they cannot be extended). Other improvements are related to the type of selectors used (in the attribute-value case, when combining selectors to C, all selectors involving an attribute already in C will be discarded).

Moreover, an efficient implementation does not test *minimality* as a final condition, but restricts the search by discarding complexes as soon as it is known that they cannot be minimal (considering rules already generated).

The exploration space of candidate rules is controlled by *stopping conditions* SC connected with requirements **a** and **b**. Requirement **c** is taken into account through the discrimination threshold d. Additionally, these conditions significantly restrict computational complexity. Here, the following stopping conditions are introduced:

Let C be the complex currently examined,

SC1: $\mid [C]_K^+ \mid < l$, where l is the smallest number of positive examples that a rule must cover (absolute strength requirement),

SC1': $\frac{\mid [C]_K^+ \mid}{\mid K \mid} < l'$, where l' is the smallest percentage of positive examples that a rule must cover (relative strength requirement),

SC2: Size$(C) > m$, where m is the largest acceptable size (length requirement).

Notice that some learning examples may not be covered by decision rules. However, this may not be damaging; it is even instructive to check the examples which are difficult to cover. Such examples can be presented to the user or expert as possible untypical cases. If they appear to be typical, it is possible to focus the search and use 'weaker' stopping conditions.

One can easily consider other stopping conditions introduced so as to satisfy additional specific user's requirements (incompatibility of some selectors, relaxing conditions when specific selectors are present in the candidate complex, ...).

6 Computational experiments

6.1 Presentation of the experiments

Our algorithm for discovery-oriented induction of 'interesting' rules was implemented as a computer program. To illustrate its usefulness, we decided to perform computational experiments on different real-life data sets.

In these experiments, a main objective was to show how to set the values of parameters (thresholds for stopping conditions) in order to derive potentially 'interesting' rules. It is clear that such values could be further refined so as to take into account more specific user's requirements. However, we believe it is extremely important to give initial values as starting points. As one will notice in experiments these values depend on each data set.

Ideally, it might be necessary to test values for all types of thresholds. It seems, however, that the most important threshold for detecting 'interesting' patterns is the strength threshold. Moreover, this threshold is clearly interrelated with the length threshold. This is why we used as a main control parameter the threshold l' involved in stopping condition SC1' (see section 5). So, for each data set considered we tested several different values of l' and observed their influence on the results. The threshold m related to stopping condition SC2 has not been really used as a control parameter. One obvious interest of this parameter from a computational viewpoint is that it guaranties that the search is polynomial. We also restricted our experiments to strictly discriminant rules ($d = 1$).

To evaluate the set of discovered rules, the following measures are taken into account:

- number of rules,
- average rule length,
- average rule strength,
- classification accuracy.

Let us precise that we are interested in discovering a limited number of relatively short and strong rules which have a reasonable classification ability.

As to this last point, it must be stressed again that unlike the case of classification systems, this criterion is treated as a secondary criterion. Classification accuracy was calculated by performing standard 10 fold cross-validation reclassification tests (see [22]). While performing reclassification tests, the matching of the testing example to condition parts of decision rules was used to predict the example classification. If the testing example matches several rules indicating different decision classes, the strongest class was chosen (as done in CN2, AQ15 or LERS systems). The case of possible non-matching is solved using the so-called VCR approach (introduced in [19]) where the decision class is chosen basing on the analysis of partly matching rules.

In order to appreciate the performance on the classification measure, we needed to use a classification-oriented induction technique. As in experiments we restricted to using selectors in the simplest form *(attribute=value)* and inducing discriminant rules only, the classification-oriented induction was performed by means of the LEM2 procedure (introduced by Grzymala-Busse in [5]) which is an effective procedure [6] giving rules in the above form. This algorithm follows a classical greedy scheme aiming at inducing a *minimum rule set* covering all examples. We used the authors' reimplementation of this procedure.

For comparison purposes, we also looked for *all discriminant rules.* This approach is sometimes used in some discovery systems for data sets of a limited size (see, e.g., approaches based on the so-called discernibility matrix [18]). This approach performing an exhaustive search in a space of possible rules leads very often to combinatorical difficulties. However, information about all rules possible to induce seem to be interesting if one wants to evaluate which part of them is really 'interesting' in a discovery-oriented perspective. In experiments *all rules* were looked for using our algorithm without any stopping condition. One can notice in Table 2 that such information could not be obtained for one of the data sets (*Election*) due to memory and time restrictions of the used computer system (although this test was performed on a powerful server SGI Power Challenge).

6.2 Analysed data sets

In the computational experiment we used 4 real life data sets of different size and characteristics. Three of them are coming from Machine Learning Database, University of California at Irvine and one (*Election*) is coming from the study described in [7]. In Table 1 the main characteristics of these data sets are presented. Two data sets were slightly modified. The Iris data set originally contains continuous-valued attributes which were discretized by means of Fayyad and Irani's method [3]. The *Voting* data set has been modified so as to eliminate missing values by removing a few attributes.

Table 1: Considered sets of learning examples

Data set	Type of data	Number of examples	Number of attributes	Number of decision classes	Cardinality of decision classes
Iris	botanical	150	4	3	50/50/50
Tic-tac-toe	games	958	9	2	626/332
Voting	political	435	13	2	267/168
Election	political	444	30	2	201/243

6.3 Results of the experiments

Using the implementation of the introduced algorithm, we tested systematically the following values for l' used in SC1$'$: 5%, 10%, 15%, 20%, 30% (i.e. these are values of minimum coverage of decision concept by the rule). For two data sets we induced rules without using SC2 (data sets *Iris, Election*), while for others the threshold m used in SC2 was set to 4 and 5 (for *Voting* and *Tic-tac-toe*, respectively). These values were chosen as a result of analysing the length of decision rules induced independently by the minimum rule set induction procedure LEM2. Information about the characteristics of rule sets obtained in experiments is presented in Table 2.

Results summarized in Table 2 show that for all data sets it is possible to indicate at least one set of 'interesting' rules. These sets consist of limited number of rules (comparable to the number of rules in the minimum set) characterized by an average strength usually about twice higher than for rules in the minimum set. Their average length is also shorter. Moreover, these sets of interesting rules give a classification accuracy nearly as good as rules obtained by LEM2 algorithm (which is a technique especially created for classification-oriented induction).

For instance, when analysing the *Iris* data set, it seems interesting to select a threshold value l' for SC1$'$ between 10% and 15%. Selecting for instance $l' = 15\%$ allows to induce a set of 20 rules, which are really stronger and a bit shorter than

Table 2: Characteristics of induced decision rules

Data set	Stopping conditions SC1′	Stopping conditions SC2	Number of rules	Average rule length [# conditions]	Average rule strength [# examples]	Classification accuracy in reclassification tests [%]
Iris	All rules		80	2.10	6.03	92.67
	5%	—	35	1.89	12.23	92.67
	10%	—	22	1.86	17.27	92
	15%	—	20	1.85	18.4	90
	20%	—	15	1.8	21.6	83.33
	25%	—	14	1.79	22.36	78.67
	30%	—	6	1.83	33.83	60.67
	Minimum rule set		23	1.91	11.0	95.33
Tic-tac-toe	All rules		2858	4.63	4.27	91.35
	5%	5	16	3	60.25	97.19
	10%	5	16	3	60.25	96.14
	15%	5	2	3	50	—
	20%	5	0	—	—	—
	30%	5	0	—	—	—
	Minimum rule set		24	3.67	40.83	98.96
Voting	All rules		1502	4.723	10.61	95.87
	5%	4	231	3.6	45.86	94.51
	10%	4	138	3.3	66.96	94.50
	15%	4	104	3.1	79.61	93.80
	20%	4	82	3.1	89.87	94.00
	25%	4	67	3.1	96.99	93.32
	30%	4	50	3.1	104.7	93.31
	40%	4	21	2.76	133.0	80.23
	Minimum rule set		26	3.69	43.77	95.87
Election	All rules		>260000	—	—	—
	10%	—	828	3.48	26.91	89.39
	15%	—	87	3.05	33.82	87.37
	20%	—	8	2.38	53.75	73.88
	25%	—	2	1.5	79	32.96
	30%	—	1	1	105	23.64
	Minimum rule set		48	3.27	21.176	89.41

Table 3: The detailed analysis of SC1′ threshold values distinguishing interesting rules from others

Data set	Stopping conditions		Number of rules	Average rule length [# conditions]	Average rule strength [# examples]	Classification accuracy in reclassification tests [%]
	SC1′	SC2				
Iris	15%	—	20	1.85	18.4	90
	18%	—	19	1.84	18.95	85.33
	20%	—	15	1.8	21.6	83.33
	Minimum rule set		23	1.91	11.0	95.33
Tic-tac-toe	2%	5	202	4.03	18.46	95.52
	3%	5	92	3.78	26.78	96.56
	4%	5	18	3	56.89	97.19
	5%	5	16	3	60.25	97.19
	10%	5	16	3	60.25	96.14
	12%	5	10	3	74.8	75.81
	14%	5	4	3	70	24.15
	15%	5	2	3	50	—
	Minimum rule set		24	3.67	40.83	98.96
Voting	30%	4	50	3.1	104.7	93.31
	32%	4	42	3.1	109.1	90.54
	35%	4	33	3	117	82.06
	40%	4	21	2.76	133.0	80.23
	Minimum rule set		26	3.69	43.77	95.87
Election	15%	—	87	3.05	37.82	87.37
	16%	—	48	2.92	40.4	85.59
	17%	—	34	2.85	42.44	81.53
	18%	—	19	2.84	48.89	80.20
	20%	—	8	2.38	53.75	73.88
	Minimum rule set		48	3.27	21.176	89.41

classification rules from the minimum rule set, while preserving a good classification accuracy. Similarly, for other data sets the following values for l' seem to give interesting rule sets: 5%–10% for *Tic-tac-toe*, 20%–30% for *Voting* and around 15% for *Election*.

It is worth noticing that the values and their range are quite different depending on the data set.

One can also notice that the option of generating *all rules* seem to be ineffective and useless from the discovery point of view. It gives an extremely high number of too long, week and specific rules.

Moreover, the detailed analysis of the influence of l' values on the 'quality' of induced rule sets show that exceeding some of the SC1′ values leads to sharp deterioration of at least one measure. In particular it refers to the number of rules and classification accuracy (see, e.g., l'=15% for *Tic-tac-toe*, 25% for *Election*). These

values depend on data sets but for all of them the observed deterioration is very sharp. This shows that we must look for threshold values allowing to obtain a balance between all relevant criteria.

In order to examine this effect more precisely we decided to focus for each data set on the critical threshold values. The obtained results are presented in table 3 and show more precisely where are the critical values distinguishing still interesting rules from others. Such a table is precious when trying to determine 'interesting rules'.

Notice also that the sets of rules are included in each other as the threshold value l' increases. This allows to consult first the strongest rules (assumed to be the most 'interesting') and if needed consult the other ones progressively.

7 Final remarks

In this paper we introduced and discussed discovery-oriented induction of decision rules. We believe that this perspective is more promising for rule induction than the classical perspective of classification. A reason is that other types of classification systems (based, e.g., on neural network or statistical approaches) often obtain at least as good results as classifiers based on induced rules. However, these alternative approaches do not compete as to the possibilities of explanation.

Discovery-oriented induction, whose purpose is to obtain 'interesting' rules, requires to take into account various criteria which contribute to define the meaning of 'interesting'. Moreover, this definition must take into account specific user's requirements. This clearly shows that approaches for discovery-oriented induction must be quite different from approaches for classification-oriented induction whose main and often unique purpose is to perform well regarding a criterion related to classification efficiency.

We presented a general algorithm which can be easily customized to take into account requirements related to the various criteria allowing to define 'interesting' rules. In addition, we showed, on the basis of several experiments, that the values of the thresholds intervening in the requirements can be defined rather easily by trying to obtain a balance between all criteria. This balance clearly depends on the importance assigned to each criterion, which must be determined in accordance with the specific needs of the user. However, we noticed in our experiments that it is possible to determine a range of values for the thresholds which leads to good results regarding all criteria. More precisely, we were always able to determine sets of rules which were significantly better than a set of classification rules considering the criteria number of rules, average rule length and average rule strength without decreasing significantly the classification accuracy.

Acknowledgements

The research has been supported by French-Polish joint research project no. 5237; moreover the first and second authors have been supported by grant no. 8 - S503 016 06 from State Committee for Scientific Research (Komitet Badan Naukowych). Other acknowledgements are directed to Poznan Supercomputer and Network Center for giving the possibility to use in experiments the computing server SGI Power Challenge. Finally, the authors would like also to thank P.M. Murphy and D.W. Aha from University of California at Irvine for providing the access to data sets from *UCI Repository of machine learning databases.*

References

[1] R. Agrawal, T. Imeliński and A. Swami. Database Mining: A Performance Perspective. *IEEE Transactions on Knowledge and Data Engineering* 8(6), pages 560—573, 1993.

[2] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3, pages 261–283, 1989.

[3] U.M. Fayad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of 13th Int. Conf. on Machine Learning*, Morgan Kaufmann, pages 1022–1027, 1993.

[4] W.J. Frawley, G. Piatetsky-Shapiro, and Christopher Matheus. Knowledge Discovery in Databases - An Overview. In G. Piatetsky-Shapiro and Christopher Matheus, editors, *Knowledge Discovery in Database,* AAAI/MIT Press, pages 1–30, 1991.

[5] J.W. Grzymala-Busse. LERS - a system for learning from examples based on rough sets. In R. Słowiński, editor, *Intelligent Decision Support*, Kluwer Academic Publishers, pages 3–18, 1992.

[6] J.W. Grzymala-Busse. Managing uncertainty in the machine learning from examples. In *Proc. of the 3rd Workshop on Intelligent Information Systems*, Wigry, Poland, IPI PAN Press, pages 70-84, 1994.

[7] M.Hadjmichael and A.Wasilewska. Rough sets-based study of voter preferences in 1988 USA presidental election. In R. Słowiński, editor, *Intelligent Decision Support*, Kluwer Academic Publishers, pages 137–152, 1992.

[8] M. Holsheimer and M. Kersten. Architectural Support for Data Mining. *CWI Technical Report* CS-R9429, CWI Amsterdam, 1994.

[9] R.S. Michalski. A theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufman, pages 83–134, 1983.

[10] R.S. Michalski. Seeking Knowledge in the Flood of Facts In *Proc. of the 3rd Workshop on Intelligent Information Systems*, Wigry, Poland, IPI PAN Press, pages 85-102, 1994.

[11] R.S. Michalski, I. Mozetic, J. Hong and N. Lavrac. The multipurpose incremental learning system AQ15 and its testing application to three medical domains. In *Proc. of the 5th Nat. Conf. on Artificial Intelligence AAAI-86*, pages 1041–1045, 1986.

[12] P.M. Murphy and D.W. Aha. UCI Repository of machine learning databases. University of California at Irvine, Dept. of Computer Science.

[13] G. Piatetsky-Shapiro. Discovery, Analysis and Presentation of Strong Rules. In G. Piatetsky-Shapiro and Christopher Matheus, editors, *Knowledge Discovery in Database* AAAI/MIT Press, pages 229–247, 1991.

[14] J.R. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1992.

[15] P. Riddle, R. Segal and O. Etzioni. Representation Design and Brute-force Induction in a Boening Manufacturing Domain. *Applied Artificial Intelligence*, 8, pages 125-147, 1994.

[16] R. Segal and O. Etzioni. Learning Decision Lists Using Homogenous Rules. In *Proceedings of the AAAI-94 Conference*, pages 619-625, 1994.

[17] J.W. Shavlik and T.G. Dietterich, editors. *Readings in Machine Learning*, Morgan Kaufman, 1990.

[18] A. Skowron. Boolean reasoning for decision rules generation. In J. Komorowski and Z.W. Ras, editors, *Methodologies for Intelligent Systems, LN in AI 689*, Springer Verlag, Berlin, pages 295–305, 1993.

[19] R. Słowiński and J.Stefanowski. Rough Classification with Valued Closeness Relation. In Diday E. et al. (editors), *New Approaches in Classification and Data Analysis*, Springer Verlag, Studies in Classification, Data Analysis and Knowledge Organization, pages 482-489, 1993.

[20] J. Stefanowski and D. Vanderpooten. A general two stage approach to rule induction from examples. In W. Ziarko, editor, *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, pages 317-325, 1994.

[21] S.M. Weiss, R.S.Galen and P.V. Tadepalli. Maximizing the predictive value of production rules, *Artificial Intelligence*, 45, pages 47-71, 1990.

[22] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*, Morgan Kaufmann, 1991.

[23] W. Ziarko and N. Shan. KDD-R: A comprehensive system for knowledge discovery in databases using rough sets. In T.Y. Lin, A.M. Wildberg, editors *Post-proceedings of the 3rd Int Workshop on Rough Sets and Soft Computing*, November 1994, San Jose, CA., Simulation Council Inc. Press, San Diego, pages 93-96, 1994.