Induction of decision rules and classification in the valued tolerance approach

Jerzy Stefanowski[1] and Alexis Tsoukiàs[2]

[1] Institute of Computing Science
Poznań University of Technology, 60-965, Poznań, Poland
Jerzy.Stefanowski@cs.put.poznan.pl
[2] LAMSADE - CNRS, Université Paris Dauphine
75775 Paris Cedex 16, France
tsoukias@lamsade.dauphine.fr

Abstract. The problem of uncertain and/or incomplete information in information tables is addressed in the paper, mainly as far as the induction of classification rules is concerned. Two rule induction algorithms are introduced, discussed and tested on a number of benchmark data sets. Moreover, two different strategies for classifying objects on the basis of induced rules are introduced.

1 Introduction

Inducing decision rules from data sets, representing sets of learning examples described by attributes, is one of the main tasks in knowledge discovery. Most of the known algorithms find rules by inductive generalisation of learning examples descriptions [2, 4]. A key issue in all such approaches is the comparison of descriptions of examples represented in a form of attribute-values vectors. Intuitively, we have to compare descriptions of examples among them in order to establish their "similarity". We also have to compare descriptions of new objects (unseen in the learning phase) to the condition parts of induced decision rules, if these rules are used for classification aims. When such attribute-value vectors are compared, it is not always the case that a crisp relation between two descriptions can be established. This is due either to the presence of incomplete descriptions or to the presence of uncertainty, imprecision and any other source of ambiguity within the descriptions. For this purpose we developed, the so called, "*valued tolerance approach*", which consists in adopting a precise version of valued similarity when multidimensional objects are compared [5–7].

Typical rule induction algorithms are based on the exploitation of crisp comparisons. In order to be able to induce rules from examples using the valued tolerance approach we need new specific procedures. Therefore, we present two different algorithms for rule induction using the valued tolerance approach. The first algorithm finds the set of all rules that can be induced from a given set of examples. The second algorithm constructs the minimal set of rules, i.e. covering the set of examples by a smallest number of rules.

The aim of this paper is experimental. Firstly, we compare on several data sets these two algorithms. A second experiment is then conducted, where we compare two different strategies for classifying new objects on the basis of rule sets induced by these algorithms. The difference between these strategies concerns the type of information used in order to classify any new object. Moreover, we examine the use of different uncertainty aggregation operators.

The paper is organised as follows. In section 2, a brief reminder of the approach based on the valued tolerance relation is given. In section 3, we introduce two different algorithms for rule induction using this approach. In section 4, we discuss various strategies that can be applied to classify new objects on the basis of induced rules. Section 5 presents the results of the computational experiments. Discussion of these results and conclusions are presented in the final section.

2 Basic concepts of valued tolerance approach

An *information table* is composed of a set of objects U described by a set of attributes A. If objects are classfied by one distinguished attribute $\{d\} \notin A$, called a *decision attribute*, we can define a *decision table* $DT = (U, A \cup \{d\})$. The decision attribute d partitions set U into decision classes denoted as $\Phi, \Psi, \ldots$. In the valued tolerance approach we assume: $\exists\ R_B(x, y) : U \times U \mapsto [0, 1]$, R_B representing a valued tolerance relation among the objects from U, established using the attributes $B \subseteq A$. R_B, which is also a fuzzy set, satisfies two properties: (1) reflexivity: $\forall x\ R_B(x, x) = 1$; (2) symmetry: $\forall x, y\ R_B(x, y) = R_B(y, x)$. We are not going to discuss in this paper how R_B is computed (for more details see [7]). We consider R_B as established.

Following rough sets theory [3] and its extension in the valued tolerance case [5–7], given a set of objects $Z \subset U$, we define as *lower approximability* of a class Φ by Z, the degree by which all objects in Z, and all objects (more or less) similar to them, are (more or less) similar to the elements in Φ. In other words, we "measure" the degree by which set Z approximates set Φ using for "similarity" the valued tolerance R_B. More formally: $\mu_{\Phi_B}(Z) = T_{z \in Z}(T_{x \in \Theta_B(z)}(I(R_B(z, x), \hat{x})))$, $\mu_{\Phi^B}(Z) = T_{z \in Z}(S_{x \in \Theta_B(z)}(T(R_B(z, x), \hat{x})))$, where: $\mu_{\Phi_B}(Z)$ is the degree for set Z to be a B-lower approximation of Φ; $\mu_{\Phi^B}(Z)$ is the degree for set Z to be a B-upper approximation of Φ; $\Theta_B(z)$ is the tolerance class of element z; T, S, I are functions representing the usual logical operators and satisfying the De Morgan law. $R_B(z, x)$ is the membership degree of x in the tolerance class of z (at the same time it is the valued tolerance relation between x and z for attribute set B); $\hat{x}$ is the membership degree of element x in the set Φ ($\hat{x} \in \{0, 1\}$).

Decision rules are represented as: $\rho_i =_{def} \ \wedge_{c_j \in B} (c_j(x) = v) \rightarrow (d = \phi)$; where $B \subseteq A$, v is the value of condition attribute $c_j \in B$, ϕ is the value of decision attribute d. In the valued tolerance approach a special *credibility degree* is associated with each rule ρ_i. We shortly remind how this degree is calculated [5]. The valued tolerance relation $s_B(x, \rho_i)$ is used in order to indicate that example x "supports" rule ρ_i, or in other words that, x is similar to some extend to the condition part of rule ρ_i on attributes B. We denote as $S(\rho_i) = \{x :$

$s_B(x,\rho_i) > 0\}$ and as $\Phi = \{x : d(x) = \phi\}$. A credibility degree for rule ρ_i is calculated as $\mu(\rho_i) = T_{x \in S(\rho_i)}(I(s_B(x,\rho_i), T_{y \in \Theta_B(x)}(I(\mu_{\Theta_B(x)}(y), \mu_\Phi(y)))))$, where: $\mu_{\Theta_B(x)}(y) = R_B(x,y)$ and $\mu_\Phi(y) \in \{0,1\}$. We quote the following result from [6]:

Proposition 1. *Consider a rule ρ_i classifying objects to a set $\Phi \subset U$ under a set of attributes B. If T, S, I satisfy the De Morgan law and R_B is a valued tolerance, the credibility $\mu(\rho_i)$ of the rule is upper bounded by the lower approximability of set Φ by the element x_k whose description (under attributes B) coincides with the condition part of the rule.*

The proof was presented in [6]. One should observe that: - the concept of rule credibility allows to fix an acceptance threshold, let's say λ, which may avoid the generation of unsafe rules; - proposition 2 allows to consider as candidates for rule generation only the examples having a sufficient high lower approximability - not smaller than λ. This reduces the rule generation cost.

3 Algorithms of rule induction

The rough sets based rule induction algorithms can be divided into two main categories [2, 4]. The first group is focused on inducing the complete set of *all rules* in the given syntax, which can be generated from the examples. The other group of algorithms is focused on *minimal set of rules*, i.e. covering the learning examples using the minimum number of rules. Inducing all rules is characterised by exponential time complexity in the worst case, while minimal sets of rules are usually generated in a heuristic way. Following this categorisation, we also present two different algorithms for rule induction using valued tolerance.

Let us suppose that the credibility threshold for the induced rules is fixed at λ. In both algorithms descriptions of objects, being completely defined by attribute-value pairs, are considered as conjunctions of elementary conditions which can be used to create condition parts of rules. According to Proposition 1 an object x is a candidate for creating a rule indicating class Φ_i if: (1) its lower approximability $\mu_\Phi(x) \geq \lambda$ (computed for completely defined attributes, where Φ is the decision class which object x belongs to); (2) the credibility of the rule using as condition part its description is also $\mu(\rho_x) \geq \lambda$. Other objects could be skipped, as they will not lead to rules with sufficient credibility.

3.1 Algorithm inducing all rules

The algorithm is based on looking for all possible reduced descriptions of candidate objects from the decision table, which lead to rules with credibility $\mu(\rho_x) \geq \lambda$. The general schema of the algorithm is presented below.

Procedure *Allrules*(DT: decision tables; var $\mathcal{R}$: set of rules);
begin $\mathcal{R} \leftarrow \emptyset$
  for $i = 1$ **to** n **do begin**{ n – number of objects in DT }
    $x \leftarrow$ *read_i-the_object*(DT);
    if not(*exist_rule*($\mathcal{R}, x$)) **then begin**

```
          μΦ(x) ← compute_lower_approximation(Φ, x); { Φ decision class of x }
          if μΦ(x) ≥ λ then { Apply Proposition 1 }
             if μ(ρx) ≥ λ then begin{ ρx decision rule created using x }
                RT ← Create_Tree_reducts(Φ, x); {find all reduced forms of ρx}
                R ← R ∪ RT end
       end
    end
end
```

Function *Create Tree Reducts* checks possible reductions of the condition part by dropping elementary conditions. Starting from one description of a candidate object, a "tree" of all admissible reduced condition parts is constructed, where each path should fulfill sufficient rule credibility and cannot be a conjunction of conditions already used in other condition parts. The induced rules are stored in a special structure and function *Exist rule* checks whether a description of object x is equal to, or is a subset of, already induced rules. The objects in decision table DT are sorted from ones having the most complete description, so "longer" candidate objects are checked the first.

3.2 Algorithm inducing minimal set of rules

This algorithm induces in a heuristic way the smallest number of rules covering all such objects from the decision table that approximate decision classes with degree $\mu_{\Phi}(x) \geq \lambda$. By objects covered by the rule we understand the objects, which are described by the same values of attributes as used in the condition part (non zero valued tolerance relation). The main idea of *MinimalCover* algorithm is inspired by techniques of linear dropping conditions used in the *LEM1* algorithm [2]. In this form of dropping, the list of all elementary conditions in the rule ρ_x is scanned from the left to the right with attempt to drop any of $(c_j = v)$ conditions, while checking whether the simplified rule does not decrease rule credibility below threshold λ - see function *Dropcondition*. In this technique, only one reduced form of a condition part is found. The order in the list of condition is determined by function *determine order conditions* on the basis of increasing number of positive examples covered by an elementary condition. So, first these conditions are dropped, which cover the smallest number of examples belonging to the decision class indicated by the rule.

```
Procedure MinimalCover(DT: decision tables; var R: set of rules);
begin    R ← ∅
   for i = 1 to n do begin
      x ← read_i-the_object(DT);
      μΦ(x) ← compute_lower_approximation(Φ, x);
      if μΦ(x) ≥ λ then
         if μ(ρx) ≥ λ then begin{ ρx decision rule created using x }
            determine_order_conditions(x, condx);
            r ← ρx;
            for j = 1 to |condx| do begin{ perform linear dropping of }
               ρy ← dropcondition(j, r); { conditions from ρx }
```

if $\mu_(\rho_y) \geq \lambda$ **then** $r \leftarrow \rho_y$; **end**
$\mathcal{R} \leftarrow \mathcal{R} \cup r$; *remove from* DT *objects* x *covered by* r; **end**
end
end.

4 Strategies for classifying new objects

Induced decision rules are the basis for classifying new or testing objects. The classification problem is to assign such objects to a decision class on the basis of their similarity/tolerance to the condition part of rules. There are two sources of uncertainty in this problem. First, the new object will be similar to a certain degree to the condition part of a given rule (due to the valued tolerance relation). Second, the rule itself has a credibility (classification is not completely sure any more). In general, the new object will be more or less similar to more than one decision rule and such rules may indicate different decision classes (with a different membership degree). In order to make a precise decision to which class the new object belongs we consider two kinds of information: (1) rule credibility and similarity/tolerance of the new object to its condition part; (2) number of objects supporting the rule, i.e. learning examples similar to condition part of the rule and belonging to the decision class indicated by the rule. The following two classification strategies are thus proposed:
Strategy A: 1. For each decision rule ρ_i in the set of induced rules, the tolerance of new object z to its condition part, $R_B(z, \rho_i)$, is calculated (where B is a set of attributes used in the condition part of ρ_i).
2. Then, the tolerance of the object z to the condition part of the rule, is aggregated with the credibility of the rule:$\mu_{\rho_i}(z) = T(R_B(z, \rho_i), \mu(\rho_i))$.
3. The membership degree of object z to decision class Φ_i is calculated on the basis of all rules $R(\Phi_i)$ - indicating Φ_i and having $\mu_{\rho_i}(z) > 0$: as $\mu_{\Phi_i}(z) = S_{\rho_i \in R(\Phi_i)}(\mu_{\rho_i}(z))$. Choose the class with the maximum membership degree.
4. If a tie occurs (the same membership for different classes), take into account information about the relative supports of rules denoted as $Supp(\rho_i)$ (it is a ratio of the number of objects supporting the rule to the total number of examples from the given decision class). For each competitive class Φ_i and its rules $R(\Phi_i)$ calculate the aggregated support as $Supp_{\Phi_i}(z) = S_{\rho_i \in R(\Phi_i)}(Supp(\rho_i))$. The object z is classified as being a member of class Φ_i with highest $Supp_{\Phi_i}(z)$.
Strategy B 1. As in strategy A.
2. As in strategy A, but $\mu_{\rho_i}(z) = T(R_B(z, \rho_i), \mu(\rho_i), Supp(\rho_i))$.
3. As in strategy A.

Practically the two strategies differ in that the first uses a lexicographic procedure in order to consider the support of a rule, while the second uses this information directly in the membership degree. A question arising at this point is the influence on the final result of the choice of the family of T, S, I operators. In this paper we consider three particular cases of T-norms:
- the min T-norm: $T(\alpha, \beta) = \min(\alpha, \beta), S(\alpha, \beta) = \max(\alpha, \beta)$;
- the product T-norm: $T(\alpha, \beta) = \alpha \cdot \beta, S(\alpha, \beta) = \alpha + \beta - \alpha \cdot \beta$;
- the Łukasiewicz T-norm: $T(\alpha, \beta) = \max(\alpha + \beta - 1, 0), S(\alpha, \beta) = \min(\alpha + \beta, 1)$;

5 Experiments

In the first part of the experiment we wanted to compare both algorithms inducing all rules and minimal cover on several data sets taking into account the following criteria: number of rules and classification accuracy. Classification accuracy was estimated by performing 10 fold cross-validation technique. We also analysed the influence of changing the number of attributes and the number of examples in data sets on the performance of both algorithms.

Table 1. The number of rules induced by both compared algorithms from Mushroom data (first number) and obtained classification accuracy [in %] for both algorithms (second number)

No. of objects	Algorithm	Number of attributes 5	7	9	11	15	21
50	Allrules	46/68	133/82	327/82	618/84	2713/90	10499/82
	MinCover	17/64	18/74	13/76	12/76	9/84	9/84
100	Allrules	69/63	206/84	482/86	1275/90	6787/86	–
	MinCover	27/57	26/72	29/75	22/80	19/83	–
250	Allrules	87/71	306/80	873/90	2895/91	18283/96	–
	MinCover	47/67	56/75	51/81	44/82	30/92	–
500	Allrules	72/51	350/80	1109/91	4875/93	–	–
	MinCover	47/48	79/77	86/86	83/85	–	–
1000	Allrules	80/51	396/80	1138/91	6596/95	–	–
	MinCover	27/48	120/77	131/86	133/87	–	–
4000	Allrules	13/8	160/36	1052/75	–	–	–
	MinCover	13/8	88/35	193/74	–	–	–
8124	Allrules	8/2	92/17	–	–	–	–
	MinCover	8/2	50/17	–	–	–	–

We used 5 real life data sets of different size and characteristics, which are coming from Machine Learning Database, University of California at Irvine [1]. The *Breast Cancer* and *Credit* data sets, which originally contained continuous-valued attributes, were discretised by means of the minimal class entropy method. Data sets contained the following ratio of missing values: *Breast cancer* - 6.14 [%], *Credit* - 2.1 [%], *Bridge* - 5.4 [%], *Hungarian* - 25.9 [%]. Moreover, the last *Mushroom* data set was artificially changed to obtain series of data sets with different number of attributes and objects. Originally, it contained 8124 objects described by 21 attributes and classified into two categories. From this data set we randomly sampled subsets containing 5, 7, 9 and 15 attributes. Then, in order to obtain data sets diversified by number of objects, we randomly created samples of data sets containing 50, 100, 250, 500, 1000 and 4000 objects. Since we wanted all such subsets of examples to contain a certain degree of missing values, we randomly introduced missing values into each data in these series (finally each data contained 20% missing values). As computational costs were high for the *All rules* algorithm, we decided to skip some combinations of highest number of attributes and highest number of objects. All computations for *Mushroom* data sets were performed with fixed threshold value of accepted rule credibility $\lambda = 0.75$. Results are summarised in the Table 1. To classify objects, we used strategy B (it gives higher accuracy).

Table 2. Classification accuracies [in %] obtained by using different classification strategies and different representations of aggregation operators; decision rules were induced by *All rules* algorithm

Data	Strategy A			Strategy B
set	min T-norm	product T-norm	Łukasiewicz T-norm	
Breast	61.56	66.93	66.79	71.26
Credit	57.53	59	63.95	65.55
Bridges	44.84	48.05	47.53	50.64
Hungarian	72.91	74.23	75.25	76.46

In the second part of the experiment we wanted to check the influence of using the two classification strategies, A and B, on the value of classification accuracy. Moreover, we wanted to examine the influence of choosing different representations of aggregation operators. We considered three particular cases presented in section 4, i.e. the min T-norm, the product T-norm and the Łukasiewicz T-norm. These experiments were performed using the four data sets *Breast Cancer, Credit Approval, Bridges, Hungarian* and with credibility threshold $\lambda = 0.9$. The first observation was that the choice of classification strategies and T operators had no significant influence in the case of the minimal cover algorithm. On the other hand, we observed such an influence when the algorithm inducing all rules was used. Thus, we summarise the experimental results for this case in Table 2.

6 Conclusions

Two algorithms for inducing classification rules from "uncertain" information tables were presented. We consider as "uncertain" the case where comparing any two objects we obtain a valued similarity (objects are more or less similar) and the induced rules are associated a credibility degree. These two algorithms, the first inducing all rules, the second a minimal set, have been tested on a number of benchmark data sets. The experimental results clearly shows that the *All rules* algorithm induces higher number of rules than *Minimal cover*. Moreover, the number of rules induced by *Minimal cover* is relatively stable, while another algorithm induces larger and larger sets of rules with increasing number of attributes. The increase of the number of objects has smaller influence on the number of rules than increasing number of attributes. The computational time is much higher for *All rules* algorithm and it exponentially grows with increasing the number of attributes. Classification accuracy is higher for decision rules generated by *All rules* algorithm. The accuracies for both algorithms usually grow with increasing number of attributes. However, the difference of accuracies between algorithms decreases with the increase of the number of objects. To sum up, both algorithms induce sets of rules having different properties. Their choice should depend on data characteristics, interest of the user and its resources.

Comparing the results of using classification strategies, first we observed that their choice had an influence on the classification accuracy only in the case of using *All rules* algorithm. For *Minimal Cover* the differences of accuracies were not significant. The results presented in Table 2 showed that better classification accuracy was obtained by using strategy B. The aggregation of similarity, credibility and rule support degrees in a lexicographic order was less efficient. The analysis of choosing particular representation of T norms to aggregate the considered degrees showed that higher classification accuracies were obtained by using either product or Łukasiesiewicz T-norms.
Acknowledgement: The research of the first author has been supported by KBN research grant no. 8 T11C 006 19.

References

1. Blake, C.L., Merz, C.J., UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
2. Grzymala-Busse J. W. LERS - A system for learning from examples based on rough sets, in Słowiński R. (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer Academic Publishers, 1992, 3–18.
3. Pawlak Z., *Rough sets. Theoretical aspects of reasoning about data.* Kluwer Acad. Publs., Dordrecht, 1991.
4. Stefanowski J., On rough set based approaches to induction of decision rules, in Polkowski L., Skowron A. (eds.), *Rough Sets in Data Mining and Knowledge Discovery*, Physica-Verlag, 1998, 500–530.
5. Stefanowski J., Tsoukiàs A., On the extension of rough sets under incomplete information, in N. Zhong, A. Skowron, S. Ohsuga, (eds.), New Directions in Rough Sets, Data Mining and Granular-Soft Computing, Springer Verlag, LNAI 1711, Berlin, 1999, 73–81.
6. Stefanowski J., Tsoukiàs A., Valued tolerance and decision rules, in W.Ziarko, Y.Yao (eds.), *Rough Sets and Current Trends in Computing*, Springer Verlag, LNAI 2005, Berlin, 2001, 212-219.
7. Stefanowski J., Tsoukiàs A., Incomplete information tables and rough classification. *Computational Intelligence*, 2001, vol. 17, 545–566.