Electronic Notes in Theoretical Computer Science 82 No. 4 (2003)
URL: http://www.elsevier.nl/locate/entcs/volume82.html 10 pages

Rough Set Approach to Pattern Extraction from Classifiers

Jan Bazan [1]

Institute of Mathematics, University of Rzeszów
Rejtana 16A, 35-959 Rzeszów, Poland

James F. Peters [2]

Department of Electrical and Computer Engineering, University of Manitoba
Winnipeg, Manitoba R3T 5V6, Canada

Andrzej Skowron, Nguyen Hung Son and Marcin Szczuka [3]

Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland

Abstract

We discuss the impact of inductive reasoning on the rough set to concept approximation. In inductive reasoning one cannot define inclusion degrees of object neighborhoods directly into the target concepts but only into some neighborhoods relevant to such concepts. Such degrees together with degrees of inclusion of patterns in target concepts make it possible to define outputs of classifiers for new classified objects. We show how among formulas used for classifier construction from decision rules one can search for new patterns relevant for the incremental concept approximation.

Key words: approximation space, concept approximation, inductive reasoning, classifiers, rough sets, patterns.

1 Introduction: Rough Sets and Inductive Reasoning

In inductive reasoning we would like to approximate concepts over a universe of objects, say U^{∞}, wider than the universe U of objects in a given decision system. In other words, assuming $U \subset U^{\infty}$, we would like to approximate concepts over U^{∞} which are extensions of decision classes in a given decision

[1] Email: bazan@univ.rzeszow.pl
[2] Email: jfpeters@ee.umanitoba.ca
[3] Email: skowron@mimuw.edu.pl, son@mimuw.edu.pl, szczuka@mimuw.edu.pl

system. In this section, we present the relevant approximation spaces for such concepts, and show how to induce *classifiers* approximating those concepts. We also discuss the relationships between the whole process and different approaches pursued in the fields like machine learning, pattern recognition, data mining and knowledge discovery [7,4,5,15,16,21].

The main observation is that, in the considered case, it is necessary to induce also a relevant approximation space. Such a space is usually different from the partition defined by the conditional attributes of a given decision system. It consists of some subsets of U^∞, called neighborhoods of objects. It should be emphasized that neighborhoods usually create a covering of U^∞, not necessarily a partition. They are defined by *patterns* chosen from some relevant *pattern languages*. In practical applications it is often necessary to specify a data model using a particular description in a pattern language. Moreover, the description usually is consistent only on a given part of the model, since the whole original model is often only partially specified. [4] In order to indicate that a given model is specified by a particular description, we use the term *description model*.

The structure of the pattern languages and the patterns themselves should be discovered. The whole process is quite complex and is illustrated in Figure 1, where:

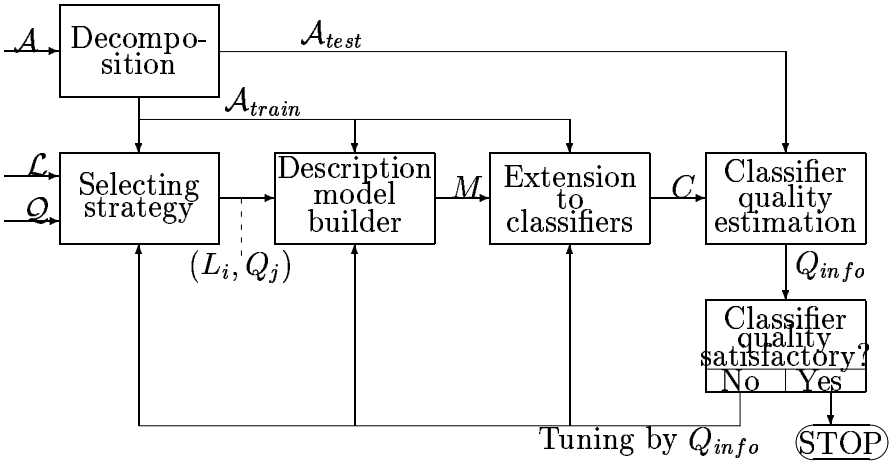

Fig. 1. Approximation space and classifier construction using rough sets.

- $\mathcal{A} = (U, A, d)$ denotes a decision system
- $\mathcal{A}_{train}$ and $\mathcal{A}_{test}$ are training and testing subsystems of $\mathcal{A}$, respectively
- $\mathcal{L} = \{L_i\}_{i \in I}$ is a family of pattern languages
- $\mathcal{Q} = \{Q_j\}_{j \in J}$ is a family of quality measures for description models
- M is a description model covering objects in U

[4] We will discuss this issue in more detail later in this section.

- C is a classifier obtained form M and covering the (almost) whole universe U^{∞}.

Elements of L_i are formulas called *patterns*. Patterns define, in a given decision system, sets of objects in which they are satisfied. Description models describe decision classes of $\mathcal{A}$, by using patterns from L_i and some inclusion measures of those patterns in decision classes. The description models can be built by means of, e.g., decision rules over descriptors from L_i.

Quality measures can be used as criteria for tuning the model. For a given L_i and Q_j, one can search for a description model using patterns from L_i which is (sub-) optimal with respect to the measure Q_j. However, the goal is to induce the relevant description model for the induced classifier, covering the whole universe of objects.

This, in particular, makes it necessary to tune parameters of the description quality measure. There are many ways to specify quality measures. For example, a measure Q_j, can be specified using the *minimum description length principle* [13,20], where one estimates the quality of approximation, as well as the size of the description model defined. The minimum description length principle requires the choice of a description of the smallest size from among those descriptions with the same approximation quality. In this case, the quality measure depends on two arguments. The first argument represents the quality of approximation (e.g., using the positive region of decision classes or entropy measure). The second argument represents the measures based on the model size. A proper balance between these two arguments is generally obtained using training data. The tuning may involve thresholds for degrees of inclusion of patterns from L_i in decision classes or for the positive region size. The use of the notion of inclusion to a satisfactory degree allows one to reduce the size of the positive region description compared with descriptions based on crisp inclusion.

The whole process, presented in Figure 1, can be viewed as a search for a relevant approximation space. As we have mentioned before, such an approximation space consists of neighborhoods of objects from U. Certainly, such an approximation space is more general than what is discussed in [11].

The induced description model should be extended to a classifier of all objects from the whole universe of objects U^{∞}, not only from U (the reader is referred, e.g., to [7,17] for the definition of classifiers). Recall that for any object to be classified, it is necessary to compute its degree of inclusion in any pattern from the description model. In the case of new objects (outside of U), these degrees can suggest conflicting decisions and, together with the degrees of pattern inclusion in decision classes, create input for a conflict resolution strategy necessary to compute the classifier output.

Next, the induced classifier is tested on objects from $\mathcal{A}_{test}$. Information Q_{info} about the classifier behavior is returned from the classifier quality estimation module. If Q_{info} shows that the classifier quality is unsatisfactory, it is used to tune parameters in different modules presented in Figure 1 and

to reconstruct the classifier to obtain a new one with a better quality. In addition, matching strategies for objects and patterns as well as parameters for conflict resolution strategy can also be tuned. The parameters involved in the tuning process can, for instance, be inclusion degree thresholds, parameters characterizing approximation quality or parameters measuring the description model size.

As a typical example, one can consider the language of patterns consisting of conjunctions of descriptors over a selected set of attributes. More complex pattern language can include conjunctions of formulas that are disjunctions of descriptor conjunctions.

1.1 Classifiers

An important class of information granules create classifiers. The classifier construction from DT can be described as follows:

(i) First, one can construct granules G_j corresponding to each particular decision $j = 1, \ldots, r$ by taking a collection $\{g_{ij} : i = 1, \ldots, k_j\}$ of left hand sides of decision rules for a given decision.

(ii) Let E be a set of elementary granules (e.g., defined by conjunction of descriptors) over $\mathcal{A} = (U, A)$. We can now consider a granule denoted by

$$Match(e, G_1, \ldots, G_r)$$

for any $e \in E$ that is a collection of coefficients ε_{ij} where $\varepsilon_{ij} = 1$ if the set of objects defined by e in $\mathcal{A}$ is included in the meaning of g_{ij} in $\mathcal{A}$, i.e., $Sem_{\mathcal{A}}(e) \subseteq Sem_{\mathcal{A}}(g_{ij})$ and 0, otherwise. Hence, the coefficient ε_{ij} is equal to 1 if and only if the granule e matches in $\mathcal{A}$ the granule g_{ij}.

(iii) Let us now denote by $Conflict\_res$ an operation (resolving conflict between decision rules recognizing elementary granules) defined on granules of the form $Match(e, G_1, \ldots, G_r)$ with values in the set of possible decisions $1, \ldots, r$. Hence,

$$Conflict\_res(Match(e, G_1, \ldots, G_r))$$

is equal to the decision predicted by the classifier

$$Conflict\_res(Match(\bullet, G_1, \ldots, G_r))$$

on the input granule e.

Hence, classifiers are special cases of information granules. Parameters to be tuned are voting strategies, matching strategies of objects against rules as well as other parameters like closeness of granules in the target granule.

The classifier construction is illustrated in Figure 2 where three sets of decision rules are presented for the decision values $1, 2, 3$, respectively. Hence, we have $r = 3$. To avoid too many indices in Figure 2, we write α_i instead of g_{i1}, β_i instead of g_{i2}, and γ_i instead of g_{i3}, respectively. Moreover, $\varepsilon_1, \varepsilon_2, \varepsilon_3$, denote

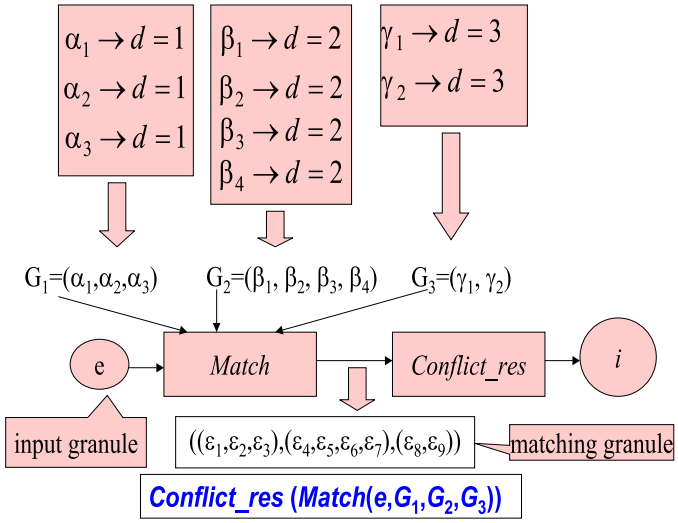

Fig. 2. Classifiers as Information Granules

$\varepsilon_{1,1}, \varepsilon_{2,1}, \varepsilon_{3,1}$; $\varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7$ denote $\varepsilon_{1,2}, \varepsilon_{2,2}, \varepsilon_{3,2}, \varepsilon_{4,2}$; and $\varepsilon_8, \varepsilon_9$ denote $\varepsilon_{1,3}, \varepsilon_{2,3}$, respectively.

The reader can now easily describe more complex classifiers by means of information granules. For example, one can consider soft instead of crisp inclusion between elementary information granules representing classified objects and the left hand sides of decision rules or soft matching between recognized objects and left hand sides of decision rules.

2 Pattern Extraction from Classifiers

We have discussed the classifier structure. In particular, we have emphasized a complex problem of classifier optimization. There are many parameters to be tuned in classifier optimization. They are related, e.g., to decision rules used to build description models of concepts on a given set of training cases, to matching strategies of objects and rules, to conflict resolution strategies resolving conflicts between decision votes of rules matched by classified objects.

We would like to illustrate a tuning process of parameters at the very end of classifier construction when conflict resolution strategy is used.

For simplicity, let us consider a case of classifier for a one concept and binary decisions $\{yes, no\}$ for such concept. The reader can easily extend our example to the case when classification is done with respect to more concepts or more decision values like *I don't know*.

In discussing description models we have observed that these models can be exact for sets of training cases (i.e., they can exactly define a concept restricted to training cases). However, if the same description models are used for classification of new cases their classification quality is unsatisfactory. Hence, we have suggested that it is necessary to search among description models for models better predisposed for classifying new objects. Quite often,

by choosing relevant models not exactly describing a given concept on the set of training cases one can finally construct classifier with high classification quality. Such description models should make it possible to construct patterns relevant for the target concept approximation, i.e., patterns which are included in concept or in its complement not only on the set of training cases but also with a high chance on sets of unseen so far, new cases. It is important to note that searching for such models is based on some inductive hypotheses, which are not necessarily satisfied for a given data set. Hence, this search process often makes it necessary to use different optimization criteria based on different such hypotheses to be able finally to discover a relevant model for a given data set. One such hypothesis can be based on the minimal description length principle.

We would like to illustrate the above idea by showing how one can extract patterns relevant for concept approximation by tuning of parameters in a conflict resolution strategy.

Assume, the values of an exemplary classifier for a given concept are based on two weights w_+ and w_-. The weights are functions of objects and use as parameters the set of (minimal) decision rules [6] derived for decisions *yes* and *no* from a given training decision table and a matching strategy. The value $w_+(x)$ for a given object x is equal to the ratio of the number of all objects satisfying the left hand sides of decision rules matched by x by the number of objects in the decision class corresponding to the decision value *yes*. More formally

$$w_+(x) = \frac{|\bigcup\{\|lh(r)\|_{\mathcal{A}_{tr}} : x \in \|lh(r)\|_{\mathcal{A}_{tr}} \text{ and } rh(r) \text{ is equal to } d = yes\}|}{|\|d = yes\|_{\mathcal{A}_{tr}}|}$$

where $lh(r), rh(r)$ denote the left and right hand side of the decision rules r, respectively; $\mathcal{A}_{tr}$ is the training decision table for the considered concept with binary decision d having the value set $V_d = \{yes, no\}$.

The weight $w_+(x)$ expresses a vote strength for *yes* for decision rules matching x. It summarizes the strength of all decision rules matching (recognizing) the object x and voting for the decision *yes*. The strength is normalized by the size of the decision class corresponding to *yes*. The weight $w_-(x)$ is defined analogously for the decision value *no*. In literature one can find many other voting strategies used in classifier construction (cf. [21]).

Our inductive hypothesis, based on intuition behind constructed weights, is the following one. One can expect that if for a given object x the value of the difference $w_+(x) - w_-(x)$ is positive but too small than the prediction of the decision *yes* in such a case will be risky because arguments *for* the decision *yes* and against the decision *yes* are almost indistinguishable. It means, that there is a high chance (even if not for the training table than for a test set consisting of new cases) that among cases with a small (absolute) value of difference $w_+(x) - w_-(x)$ are cases with the real decision *yes* and cases with the real decision *no*.

In the simplest case, if, e.g., $w_+(x) > 0$ and $w_-(x) = 0$ then one can

predict the decision value *yes*. Similarly for the decision value *no*. In case when both weights for an object are positive one can classify object to the boundary region. However, using such a strategy we usually obtain a large boundary region. Hence, one can look for more sophisticated strategies for resolving conflict between weights. Below we present an example of such a strategy assuming that both weights are non-negative.

One can choose an optimization strategy based on two positive thresholds t_1, t_2 and search for as small as possible values of them such that if $w_+(x) - w_-(x) \geq t_1$ or $w_-(x) - w_+(x) \geq t_2$ then taking the decision *yes* or *no*, respectively, on the basis of the difference of weights is not risky. In other words the patterns defined by constraints $w_+(x) - w_-(x) \geq t_1$ or $w_-(x) - w_+(x) \geq t_2$ are relevant, i.e., sets of objects satisfying them are included in the decision class corresponding to *yes* and *no*, respectively. Optimization of t_1, t_2 can be treated as a minimization of the boundary region (defined by c).

Now one can consider an exemplary classifier c as a condition attribute with two parameters t_1, t_2 and the value set $V_c = \{0, 1, ?, don't\_know\}$ defined by

$$c(x) = \begin{cases} don't\_know & \text{if } w_+(x) = w_-(x) = 0 \\ 1 & \text{if } w_+(x) - w_-(x) \geq t_1 \\ 0 & \text{if } w_-(x) - w_+(x) \geq t_2 \\ ? & \text{if } -t_2 < w_+(x) - w_-(x) < t_1 \end{cases}$$

It is necessary to choose some criteria for tuning of the parameters t_1, t_2. For example, one can search for parameters t_1, t_2 such that patterns defined by the descriptors $c = 1$ and $c = 0$ are supported by as large as possible number of objects and are included in the decision classes defined by *yes* and *no*, respectively. In this way, the boundary region defined by the descriptor $c =?$ is minimized and at the same time the lower approximations for the concept *small* and its complement are maximized.

The approximations of the concept are calculated on the set of objects from a given decision table for which the value of the attribute c is different from *don't_know*.

In general, on new cases the constructed patterns will be not exactly included in the decision classes but usually we are satisfied if they will be included up to satisfactory degree. This requires a modification of the concept approximations assuming a given pattern is included in the lower approximation of a given concept if it is included in the concept up to a satisfactory degree.

The misclassification property of the classifier can be illustrated using, e.g., a confusion matrix. Such matrix represent a report on the classification quality of a given classifier on a given data table (sample). An example of

confusion matrix is presented below:

```
CONFUSION MATRIX:
                Predicted
              yes   no  yes_OR_no  Not_covered
Actual  yes 119   6       83        17
        no  10    155     92        18
```

Columns in the matrix describe classifier prediction. In the considered example $119 + 10 = 129$ cases have been classified for the decision class with the decision value *yes*; $155 + 6 = 161$ cases for the class with the decision value *no*; $83 + 92 = 175$ cases have been classified for the boundary region and $17 + 18 = 35$ objects have not been recognized on a given sample by the classifier (i.e., they are not matched by any rule). The rows describe the quality of prediction. The real value *yes* in a given sample (data table) was for $119+6+83+17 = 225$ objects. Out of them 119 have been properly classified by the classifier, for 6 of them the classifier predicted the decision *no* (while the correct decision was *yes*); 83 objects with the real decision *yes* have been classified to the boundary region by the classifier and 17 objects with the real decision *yes* have not been recognized by the classifier. The real value *no* in a given sample (data table) was for $10+155+92+18 = 275$ objects. Among them 155 have been correctly classified by the classifier; 10 of them have been classified incorrectly; 92 objects have been classified to the boundary region and 18 objects have not been recognized by the classifier.

Now one can ask how to approximate incrementally the considered concept on the set of objects extended by a given set of testing objects for which the confusion matrix has been constructed. This is in a sense a posteriori approximation, i.e., approximation of the concept on the union of the training and testing sets. Such approximation can be based on parameterized patterns defined by classifiers, e.g., by means of weights used for expressing different voting strategies between rules. By tuning the parameters one can search for relevant patterns, i.e., patterns included to sufficiently high degrees in the concept or its complement. Observe that new patterns can be related to many different classifiers. Hence, the approach is also related to strategies for ensembles of classifiers [3]. Strategies used for conflict resolution in classifiers can help to induce new relevant patterns from which new strong decision rules can be obtained. These patterns together with those defined by decision rules generated from the training set are used for the rough set concept approximation on the extended universe of objects.

3 Conclusions

We have outlined the rough set approach for incremental learning of concept approximation. It was stressed that in inductive reasoning one cannot define

inclusion degrees of object neighborhoods directly into the target concepts but only into some relevant to such concepts patterns (e.g., left hand sides of decision rules) (see, e.g., [23,18,1]). Such degrees together with degrees of inclusion of patterns in target concepts make it possible to define outputs of classifiers for new classified objects. We have shown that some expressions over which classifiers are constructed define new patterns relevant to the concept approximation on the extension of the training object set by testing objects [1].

Our approach is different from [14,19,22]. We propose to search for relevant patterns in language defined by conflict resolution strategies rather than by tuning the existing rules. One of the possible extension of our work will be to develop a method combining both approaches.

The next step of our project will be to verify the approach on different data sets.

Acknowledgements

The research of Jan Bazan, Hung Son Nguyen, Andrzej Skowron and Marcin Szczuka has been supported by the State Committee for Scientific Research of the Republic of Poland (KBN) research grant 8 T11C 025 19 and by the Wallenberg Foundation grant. The research of James Peters has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) research grant 185986.

References

[1] Bazan, J., Nguyen, H.S., Skowron, A., Szczuka, M.: A view on rough concept approximations (to appear).

[2] Brown, F.M.: *Boolean Reasoning.* Kluwer Academic Publishers, Dordrecht, 1990.

[3] Dietterich. D.T.: Machine learning research. Four current directions. AI Magazine **18**(4) 1997, pp. 97–136.

[4] Friedman, J.H., Hastie, T., Tibshirani, R.: *The Elements of Statistical Learning.* Springer-Verlag, Heidelberg, 2001.

[5] Kloesgen, W., Żytkow, J. (eds.), *Handbook of KDD*, Oxford University Press, 2002,

[6] Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In: *[9]*, pp. 3–98, 1999.

[7] Mitchell, T.M.: *Machine Learning.* McGraw-Hill, New York, 1997.

[8] Pal, S.K., Polkowski, L., Skowron, A. (eds.): *Rough-Neuro Computing: Techniques for Computing with Words.* Springer-Verlag, Berlin, 2003. (to appear).

[9] Pal, S.K., Skowron, A. (eds.): *Rough Fuzzy Hybridization: A New Trend in Decision–Making.* Springer-Verlag, Singapore, 1999.

[10] Pawlak, Z.: Rough sets. *International Journal of Computer and Information Sciences* **11**, 1982, pp. 341–356.

[11] Pawlak, Z.: *Rough Sets. Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht, 1991.

[12] Polkowski, L., Skowron, A. (eds.): *Rough Sets in Knowledge Discovery* **1-2**. Physica-Verlag, Heidelberg, 1998.

[13] Rissanen, J.J.: Modeling by shortest data description, *Automatica* **14**, 1978, pp. 465-471.

[14] Shan, N., Ziarko, W.: Data-based acquisition and incremental modification od decision rules. *Computatioinal Intelligence* **11**(2), 1005, pp. 357–370.

[15] Skowron, A.: Rough sets in KDD. In: Z. Shi, B. Faltings, and M. Musen (eds.), *16-th World Computer Congress (IFIP'2000): Proceedings of Conference on Intelligent Information Processing (IIP'2000)*, Publishing House of Electronic Industry, Beijing, 2000, pp. 1–17.

[16] Skowron, A., Pawlak, Z., Komorowski, J., Polkowski, L.: A rough set perspective on data and knowledge. In: W. Kloesgen, J. Żytkow (eds.), *Handbook of KDD*, Oxford University Press, 2002, pp. 134–149.

[17] Skowron, A., Stepaniuk, J.: Information granules and rough-neuro computing. (to appear in [8]).

[18] Skowron A., Szczuka M.: Approximate reasoning schemes: Classifiers for computing with words. Proceedings of SMPS 2002, Advances in Soft Computing series, Physica Verlag, Heidelberg, 2002, pp. 338–345.

[19] Susmaga, R.: Experiments in incremental computation of reducts. In [12] **1**, pp. 500-529.

[20] Ślęzak, D.: *Approximate Decision Reducts.* Ph.D. Thesis, Warsaw University, 2002 (in Polish).

[21] Watanabe S.: Pattern Recognition: Human and Mechanical, Wiley, Toronto, 1985

[22] Wojna, A.: Constraint based incremental learning of classification rules. LNAI **2005** Springer-Verlag, 2001, pp. 428–435.

[23] Wróblewski, J.: *Adaptive Methods of Object Classification.* Ph.D. Thesis, Warsaw University, 2002 (in Polish).